Les technologies du web appliquées aux données structurées

Emmanuelle Bermès, Gautier Poupeau

Les technologies du web appliquées aux données structurées

Emmanuelle Bermès et Gautier Poupeau

Diplômée de l'École nationale des chartes et de l'ENSSIB, Emmanuelle Bermès est actuellement chef du service multimédia au Centre Pompidou, où elle pilote notamment le projet de Centre Pompidou Virtuel. Conservateur à la Bibliothèque nationale de France entre 2003 et 2011, elle a travaillé sur Gallica, sur le projet de préservation numérique SPAR et sur l'évolution des catalogues vers le web sémantique. Elle est active dans des réseaux internationaux comme l'IFLA, Europeana, et le W3C où elle a co-présidé un groupe de travail sur le web de données et les bibliothèques. manue@figoblog.org

Titulaire d'un DEA en sciences de l'information de l'École nationale supérieure des sciences de l'information et des bibliothèques (ENSSIB), Gautier Poupeau est consultant-architecte des données chez Antidot depuis juillet 2010. Après avoir géré le site web et les éditions électroniques de l'École nationale des chartes, il a été consultant chez Unilog puis Atos Origin. Il est l'auteur du blog Les petites cases. www.lespetitescases.net, gpoupeau@antidot.net

Défini par le World Wide Web Consortium (W3C) depuis la fin des années 1990, le web sémantique est un ensemble de standards et de technologies qui vise à faire entrer les données structurées dans l'environnement du web, en adoptant les principes, l'architecture et les techniques qui ont permis la construction de cet espace d'interopérabilité globale qu'est la toile d'aujourd'hui.

Ces technologies ouvrent des perspectives essentielles pour les institutions culturelles et scientifiques qui, de par leur activité, sont naturellement détentrices ou productrices de quantités importantes de données structurées. Il s'agit pour ces institutions non plus de développer des standards spécifiques à un domaine métier en particulier, ou de mettre en place des passerelles qui restent déterminées principalement par la nature culturelle ou scientifique des objets concernés, mais d'entrer dans un écosystème beaucoup plus global qui leur permettra d'interagir avec des acteurs de toute nature, qu'ils soient publics ou privés, culturels ou industriels, politiques ou économiques, individuels ou collectifs. Des enjeux comme l'interopérabilité, l'ouverture et la réutilisation des données publiques se trouvent à la clef de cette évolution.

L'objectif de ce chapitre est de montrer les possibilités nouvelles offertes par les technologies du web sémantique quant au traitement et à l'exploitation des données structurées. Nous définirons les principales briques qui constituent cet environnement technologique : les URI, RDF, OWL, SPARQL, RDFa..., et enfin présenterons leur application la plus directe, le web de données.

1 Adopter l'architecture du web

Le web tel que nous le connaissons aujourd'hui fait à ce point partie de notre quotidien que nous oublions qu'il constitue un espace et un modèle d'interopérabilité unique en son genre dans le domaine de l'informatique. Accessible sur toutes sortes de terminaux, de l'ordinateur de bureau au téléphone portable, à travers une variété de logiciels développés par des entreprises concurrentes sur des systèmes d'exploitation incompatibles entre eux, le web fournit à l'utilisateur une expérience essentielle, celle de la mise à disposition d'un espace documentaire global et cohérent, où les frontières entre les institutions, les pays, les entreprises ne sont plus des frontières technologiques.

Lorsque l'on parle de web sémantique, c'est d'abord l'idée de web qui est clef : en adopter l'architecture pour exprimer et échanger des informations en bénéficiant de sa qualité unique d'interopérabilité. Nous étudions ici les fondements de l'architecture du web [15] avant de voir comment elle s'applique dans l'espace documentaire puis dans celui de l'information structurée.

1.1 Les principes du web

À l'origine, l'objectif du web était de *lier et partager*, parmi un ensemble de machines connectées en réseau, *des documents lisibles pour les humains*. Le besoin initial de Tim Berners Lee, lorsqu'il élabora l'idée du web, était en effet d'offrir un espace interopérable pour partager les documents conservés par les chercheurs du CERN[1] sur leurs machines. Cet espace de partage documentaire se devait d'être décentralisé pour répondre aux impératifs de passage à l'échelle, et tenir la charge par rapport aux millions de documents concernés.

Le web, tel qu'il a été conçu pour répondre à ce besoin initial, se base sur trois principes :

[1] Centre européen de recherche nucléaire

- la mise au point du protocole de communication HTTP[2] reposant sur Internet pour mettre les machines en réseau ;
- la définition d'un système d'identifiants, qui permet de localiser de manière uniforme des ressources sur différentes machines distantes ;
- l'utilisation du principe de l'hypertexte pour relier les ressources.

1.1.1 Un réseau de machines décentralisées

Le web est une architecture décentralisée car elle ne s'appuie pas sur un seul serveur principal dont dépendrait l'ensemble du réseau, mais sur plusieurs serveurs répartis. La disparition d'un serveur ou l'ajout d'une nouvelle machine ne déstabilise pas l'ensemble du réseau. C'est la première condition de la construction du web « par le bas » (*bottom-up*) : il se construit par l'agrégation des informations qui sont publiées, et non suivant un plan prédéfini.

La mise en réseau nécessite la mise en place d'un code pour que les machines puissent se *transmettre* les informations. En ce qui concerne le web, les conditions de mise en réseau des machines (routage, transport) sont définies par la brique technologique de base de l'Internet : le protocole TCP/IP[3]. Le web est l'une des applications de cette technologie, à travers le protocole HTTP qui constitue le moyen de faire transiter le message entre deux machines. Ainsi, HTTP constitue l'une des couches applicatives d'Internet, au même titre que les protocoles POP ou IMAP pour le courrier électronique ou le fameux P2P pour l'échange de fichiers.

Le protocole HTTP définit quatre formes principales d'interaction ou méthodes entre une application cliente et le serveur :
- le verbe GET pour obtenir un contenu stocké sur le serveur ;
- le verbe POST pour soumettre au serveur des données sur un contenu en vue de leur traitement (modification ou création) ;
- le verbe PUT pour remplacer ou ajouter un nouveau contenu sur le serveur ;
- le verbe DELETE pour supprimer un contenu sur le serveur.

1.1.2 Les fondements de l'architecture du web

Le web repose sur l'idée de la séparation entre une entité abstraite et sa représentation matérielle sous la forme d'un flux. Cette dichotomie sert à offrir plusieurs manières de représenter une même entité en fonction de différents critères : la langue, le pays d'origine de l'utilisateur, la forme de la représentation, etc. Ainsi, l'architecture du web s'appuie sur trois notions fondatrices : la notion d'identifiant, celle de représentation et celle de ressource.

• *L'identifiant* conditionne pour les machines l'existence de l'entité sur le réseau. À chaque entité correspond donc un identifiant, une URI[4]. Cet identifiant permet aux machines de faire référence à une ressource, de l'identifier et de la localiser.

• *La représentation* désigne un flux fini dans un format précis. Une URI peut avoir une ou plusieurs représentations en fonction de critères tels que le format que peut traiter le client qui exécute la requête, l'encodage de caractères, la provenance géographique de la requête, la langue, etc. Dans la très grande majorité des cas, une URI n'a qu'une seule représentation, celle renvoyée par défaut par le serveur. Même lorsqu'il existe plusieurs représentations, le choix est généralement transparent pour l'utilisateur humain, car il est pris en charge par le navigateur qui sollicite par défaut le format le plus approprié.

• *La ressource*, enfin, désigne l'entité elle-même d'un point de vue conceptuel, identifiée par une URI qui possède une ou plusieurs représentations. Toute entité identifiée par une URI est donc une ressource. La ressource est constituée de la somme de l'identifiant et de toutes ses représentations.

1.1.3 L'hypertexte

Le choix d'une architecture décentralisée implique de pouvoir passer d'un document sur une machine A à un document sur une machine B. Pour cela, le web s'appuie sur le principe de l'hypertexte : relier les documents entre eux à l'aide d'un pointeur, en l'occurrence un identifiant. Toutefois, ce principe ne pouvait pas suivre des règles hiérarchiques, qui prévalaient auparavant dans l'échange de documents. Il n'était en effet pas possible de définir une organisation globale du plus général au plus spécifique ; il n'existe pas de point central ou de racine à laquelle tous les documents pourraient se rattacher. De la même manière qu'on peut ajouter des serveurs sans déstabiliser l'infrastructure, on peut ajouter des pages sans déstabiliser le graphe global que composent les pages reliées entre elles.

Par ailleurs, une approche centralisée aurait imposé de connaître à l'avance tous les documents dans le monde et leur identifiant, ce qui était impossible. La grande révolution qu'apporte le web par rapport aux autres systèmes hypertextuels de l'époque réside dans le fait qu'on peut créer le lien sans que le document cible existe. On n'impose pas la vérification de son existence, on n'a donc pas besoin de centraliser un répertoire des pages existantes.

Le besoin de pointeurs pour élaborer le réseau hypertexte impose ensuite la nécessité de disposer d'*un moyen unique de localisation et d'identification* des documents sur des serveurs distants, qui est fourni par le principe des URL[5]. Ce

---

[2] Hypertext Transfer Protocol

[3] Transmission Control Protocol/Internet Protocol

[4] Uniform resource identifier

[5] Uniform resource locator

mécanisme s'appuie à l'origine sur la localisation de la machine et du document sur la machine pour construire l'identifiant du document.

En réalité, les URL ne constituent que l'un des modes d'identification possibles au sein d'un spectre plus large, celui des URIs [voir la section 2].

1.1.4 Des standards ouverts et libres

Enfin, la caractéristique sans doute la plus importante du dispositif réside dans le fait que toute l'architecture est ouverte et repose sur des standards qui sont mis à la portée de tous. Ce qui a fait la valeur ajoutée de cette technologie et a motivé son adoption quasi universelle, c'est sa globalité et son interopérabilité. Ainsi, le web est avant tout un ensemble de standards qui permettent la dissémination de technologies partagées par tous et indépendantes des environnements matériels et logiciels. Son architecture s'appuie sur une infrastructure technique qui garantit l'interopérabilité entre des machines distribuées. Il permet de construire *un espace global d'information*, utilisant les liens pour permettre de naviguer de manière transparente d'une ressource à une autre et d'une machine à une autre.

1.2 L'architecture du web appliquée aux documents

À l'origine, les principes de l'architecture que nous venons de décrire sont appliqués à des informations documentaires échangées entre des machines sous la forme de pages web, et interprétées directement par les utilisateurs via un logiciel spécifique, le navigateur.

1.2.1 Encoder le message

Pour atteindre cet objectif, il est nécessaire de disposer d'un formalisme unique permettant d'*interpréter la structure du message*, c'est-à-dire de partager le même format d'encodage. Au niveau de la sérialisation, c'est-à-dire des modalités d'écriture, que l'on pourrait comparer avec l'alphabet utilisé pour exprimer une langue particulière à l'écrit, c'est le format SGML[6] qui est choisi à l'époque. En ce qui concerne la structure, c'est-à-dire la grammaire et le vocabulaire, qui se confondent dans le monde des langages à balise, elle est définie par le HTML[7]. À l'origine, HTML est conçu comme une DTD[8] de SGML, qui fournit une sémantique pour pouvoir encoder les documents : le nom des éléments qui les constituent, leurs fonctions et les règles d'agencement qui régissent leur usage.

1.2.2 Interpréter le code et représenter

L'outil le plus commun pour interpréter le code HTML est *le navigateur web*. C'est lui qui *prend en charge l'interprétation des éléments de structuration* qui lui sont fournis, pour transformer l'information brute en *une représentation lisible pour les humains* avec les éléments de mise en forme associés (découpage du texte en titres, paragraphes, etc., auxquels des éléments de présentation typographique peuvent être associés par l'intermédiaire d'une feuille de style, éléments imbriqués tels que des images, des vidéos, des applications, etc.).

En réalité, le navigateur n'interprète pas le message. C'est la structure du message qui est encodée en HTML, et le navigateur se contente d'en fournir une représentation sous forme de page web dont le contenu sera ensuite interprété par des humains. Ainsi, le navigateur garantit l'interaction de l'utilisateur avec le document pour pouvoir naviguer dans l'espace global d'information, mais *sans contribuer à l'interprétation du contenu du message*, au-delà des conventions d'affichage notamment typographique : par exemple, si un humain voit un élément en italique à l'intérieur d'un texte, c'est à lui d'interpréter la raison pour laquelle l'italique a été utilisé. S'agit-il d'un titre cité, d'une simple emphase ? En HTML la nature de cette information n'est pas exprimée.

1.2.3 Relier les documents

Il en va de même pour les liens hypertextes qui existent entre les documents. Le mécanisme des liens entre les pages web ne constitue rien d'autre qu'*un système de pointeurs* qui permettent d'actionner un cheminement d'une page à une autre. *Aucune précision n'est fournie en HTML sur la nature des liens qui relient deux éléments*, et c'est encore aux humains qu'il revient de l'interpréter.

De plus, comme les URIs fournissent un système d'identification global, la machine ne perçoit aucune différence entre un lien vers une ressource sur le même serveur et un lien vers une ressource distante : son comportement est exactement le même. Par exemple, lorsqu'une institution A propose sur son site un lien vers le site d'une institution B, les mécanismes permettant de savoir qu'on a changé d'institution relèvent du graphisme et de l'ergonomie (une bonne pratique consiste à identifier clairement par des artifices graphiques les liens « sortants ») ainsi que de l'analyse de l'utilisateur, qui se rendra compte qu'il a changé de « site » (la notion de site web ne correspondant en fait à aucune réalité technique).

Quant à la nature de ce lien, ou la raison profonde qui a conduit l'institution A à pointer vers l'institution B, c'est l'intelligence de l'utilisateur qui devra l'analyser : ces deux institutions sont-elles dépendantes l'une de l'autre ?

---

[6] Standard Generalized Markup Language

[7] Hypertext Markup Language

[8] Document type definition

Présentent-elles des contenus pertinents et complémentaires ? Le lien hypertexte de base ne fournit pas cette information.

1.3 Du web de documents au web de données

Dans le contexte qui nous intéresse, il s'agit d'utiliser cette architecture non pas pour relier des documents, mais pour exprimer et échanger des informations sur *des entités qui ne sont pas forcément de nature documentaire* (par exemple, des personnes, des concepts, des lieux, etc.). Alors que le principe du web documentaire consistait simplement à transporter une information de façon à la remettre à des humains, à charge pour eux de l'interpréter, celui du web sémantique est de *permettre aux machines de traiter directement* le sens contenu dans les informations qui transitent sur le réseau concernant les entités identifiées.

L'objectif est d'étendre les modalités de l'interopérabilité du web à des informations qui sont habituellement stockées en dehors de cette architecture : soit elles sont présentes dans les documents, mais sous une forme qui n'est pas directement interprétable pour des machines, soit elles sont stockées dans des bases de données et ne sont pas formalisées suivant les principes de l'architecture du web.

Dans le cas des informations stockées dans des bases de données et accessibles via un formulaire de recherche, celui-ci constitue une barrière que les machines ne peuvent franchir : il est nécessaire qu'un acteur humain formule une requête pour accéder aux informations. Dès lors, ces informations se trouvent dans ce qu'on a pu appeler le « web profond » ou « web caché » : une zone à laquelle il n'est pas possible d'avoir accès simplement en actionnant des liens.

Pour rendre ces informations accessibles à des machines, il est nécessaire de les exprimer suivant un formalisme conforme à l'architecture du web, qui rende perceptible pour les machines la nature de la relation entre les entités décrites dans la base : ce qui constitue l'essence de la donnée. Lorsqu'on parle de « données liées » (traduction littérale de *linked data*) ou de web de données, on désigne donc le fait d'appliquer les principes de l'architecture du web à la structuration des données elles-mêmes.

Cela implique, comme c'était le cas pour HTML, de formaliser le message grâce à une structure adaptée, mais qui dépasse cette fois la simple description de la mise en forme de l'information dans un document pour s'attacher à décrire les entités elles-mêmes :
- exprimer la nature de ces entités et de la relation qui les unit, nature qui pourra ensuite être partagée grâce à une sémantique commune ;
- exploiter les informations obtenues, en parcourant les liens qui existent entre les entités ou en effectuant des requêtes ;
- enfin fournir des méthodes d'écriture (sérialisation) et d'échange (protocoles) qui permettront la manipulation effective de ces informations par les machines.

Afin de concrétiser cette vision, le W3C a développé un ensemble de standards qui sont habituellement représentés dans un schéma appelé le *semantic web stack* [figure 1]. On y retrouve dans les couches basses les éléments communs que sont les URIs et XML[9] ; puis viennent s'ajouter d'autres standards (RDF, RDFS, OWL, SPARQL, etc.), cette fois dédiés à l'usage particulier de structuration et d'exploitation des données sur le web.

L'identification des entités non documentaires constitue le socle du web sémantique : c'est elle qui va permettre d'émettre des assertions concernant ces entités, de les échanger, et de les rendre interopérables pour les machines grâce à un formalisme de structuration adapté. C'est donc à cette problématique d'identification qu'il faut s'intéresser en tout premier lieu, avant d'entrer plus en détail dans les standards techniques du web sémantique.

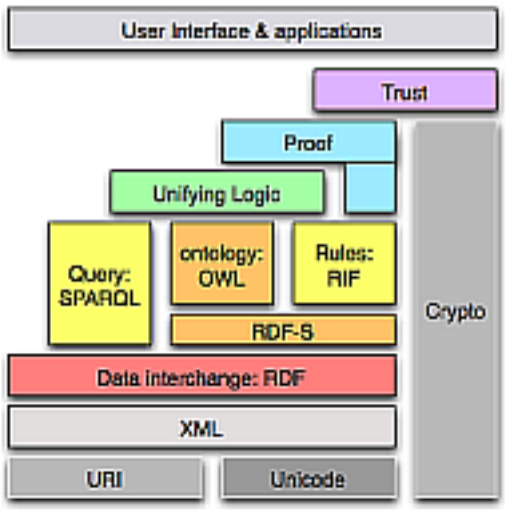

Figure 1 – Représentation schématique des technologies du web sémantique ou *semantic web stack*

[9] eXtensible Markup Language

2 Identifier : l'attribution des URIs

La première question à se poser à l'heure d'adopter les standards du web sémantique pour exprimer des données est donc celle de l'identification des entités que l'on souhaite manipuler.

2.1 Être ou ne pas être (identifié) ?

Il convient d'étudier les modalités et les impacts du choix des identifiants dans le contexte d'un projet particulier. Cette étape est essentielle dans la mesure où elle constitue l'opportunité d'*analyser les différentes entités qu'on manipule et qui doivent être identifiées*. Il faut définir, dans l'ensemble des ressources sur lesquelles on travaille, quelles sont celles qui feront l'objet d'assertions, seront reliées à d'autres, ou seront réutilisées de quelque manière que ce soit.

Ces ressources peuvent inclure les entités qui constituent le sujet principal des données sur lesquelles on travaille (des livres dans une bibliothèque, des œuvres et des artistes dans un musée, des clients et des comptes dans une banque, etc.), mais aussi toute autre ressource liée sur laquelle on pourrait avoir besoin de fournir des informations complémentaires (des lieux, des événements, des périodes historiques, des concepts, etc.).

Le fait de placer ces entités dans le web implique d'adopter pour les représenter l'architecture de celui-ci, c'est-à-dire dans un premier temps de leur donner des identifiants pour les rendre utilisables et localisables sur le réseau. Pour cela, on utilise une structure d'identifiant spécifique : les URIs [16]. Le nommage de ces entités avec des URIs fonde leur existence sur le web. L'utilisation des mécanismes de base de cette architecture, HTTP et les URIs, assure l'identification et la localisation de ces entités sur le réseau : elles deviennent des objets tangibles pour les machines.

Ce mécanisme permet d'éviter les ambiguïtés inhérentes au langage humain, puisqu'une URI ne peut identifier qu'une seule ressource au sein d'un réseau (qu'il soit physique comme le web ou virtuel).

2.2 La syntaxe des identifiants

Une fois définis les objets à identifier, il est nécessaire de définir la syntaxe qui sera utilisée pour constituer les URIs.

L'URI est un système d'identifiant dans un réseau physique ou virtuel dont la syntaxe est normalisée [figure 2]. Le *scheme* est un préfixe qui indique le contexte dans lequel les identifiants sont attribués. L'*authority* désigne une autorité dite nommante en charge d'attribuer des noms pour ce *scheme*. Le *path* est un chemin ou un nom attribué par l'autorité nommante.

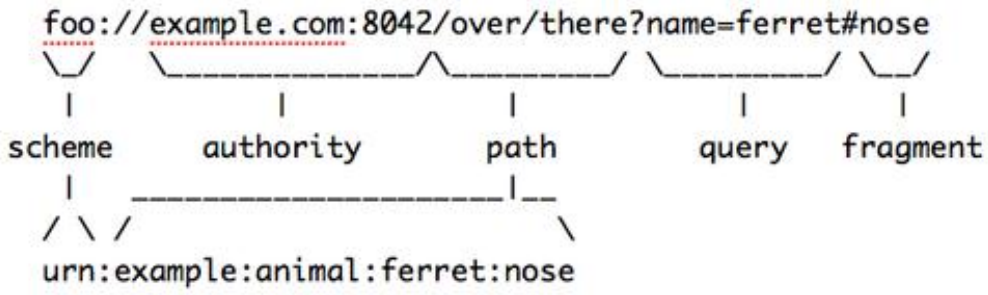

Figure 2 – Syntaxe des URIs

Appliquée au web, cette syntaxe impose l'utilisation du *scheme* « http: » qui renvoie à l'utilisation du protocole correspondant, et l'usage d'un nom de domaine agréé par l'organisme qui régit ce *scheme*, l'ICANN[10]. Dans le cas d'une URL, le *path* est constitué d'informations qui permettent d'identifier la ressource via le moyen d'y accéder (localisation physique ou mécanisme logiciel permettant d'appeler la ressource sur le réseau).

Comme nous l'avons vu, une URL est un type particulier d'URI (parce qu'elle en respecte la syntaxe) qui utilise le *scheme* « http: » et identifie une ressource principalement par le mécanisme qui permet d'y accéder. Ce mécanisme peut être le nom du fichier et son emplacement sur le serveur ; il peut être aussi une série de paramètres qui appellent une base de données, via un logiciel, ou une chaîne de caractères qui va être interprétée grâce à un annuaire permettant de déterminer l'emplacement physique de la ressource en question.

Il est syntaxiquement possible de définir des URIs qui ne seraient pas des URL (qui n'utiliseraient pas le *scheme* « http: », mais un autre *scheme* correspondant à un autre réseau, physique ou virtuel). Toutefois, un tel choix conduirait à s'exclure de l'architecture du web, et présente donc peu d'intérêt [voir la section 2.3 ci-dessous].

[10] Internet Corporation for Assigned Names and Numbers. http://www.icann.org

C'est donc dans le contexte des URL que nous proposons de travailler. Dès lors, la définition des identifiants implique le choix d'un nom de domaine et celui d'un formalisme pour les caractéristiques qui permettent d'identifier chaque ressource au sein de ce domaine. Ces choix sont guidés essentiellement par la problématique de l'unicité et de la pérennité.

La définition du nom de domaine devrait *prendre en compte les risques liés à l'obsolescence* potentielle des termes qui le composent. Par exemple, une institution telle qu'un ministère qui change souvent de nom ne devrait pas utiliser comme nom de domaine son acronyme (MCC pour le Ministère de la Culture et de la Communication), mais un terme ou un ensemble de termes générique susceptible de rester pérenne dans le temps (culture.gouv.fr).

En ce qui concerne l'identification de chaque ressource, tous *les éléments qui sont susceptibles de changer doivent être absolument exclus*, et en particulier :
- tout élément afférent à la localisation de la ressource (sur un serveur...) ;
- tout élément lié au contexte logiciel d'accès à la ressource à un moment donné (paramètres techniques tels que des identifiants de session...).

Des mécanismes de redirection ou de réécriture d'URL peuvent être utilisés pour traduire les identifiants dans un formalisme permettant l'accès à la ressource.

De fait, on va le plus souvent utiliser des éléments qui font déjà partie de la description de la ressource pour composer l'identifiant. Il s'agit de *réutiliser des identifiants existants* dont l'unicité est déjà garantie dans un contexte donné (identifiants universels tels que l'ISBN pour le livre, identifiants institutionnels tels que les cotes d'une bibliothèque, identifiants techniques tels que les clés uniques attribuées aux entrées d'une base de données).

Si aucun identifiant ne préexiste, il faut en définir de nouveaux. Deux options se présentent alors :

- définition d'identifiants opaques (des suites de caractères alphanumériques) qui ne donnent aucune information sur le contenu de la ressource ;
- définition d'identifiants signifiants qui s'appuient sur un champ différenciant (par exemple, le nom d'une personne, le titre d'une œuvre, etc.).

L'avantage des identifiants opaques réside dans la facilité de les rendre uniques et pérennes. Totalement dissociés du contenu de la ressource, ils ne sont pas sensibles à des événements qui pourraient affecter celle-ci au cours du temps (par exemple, un changement de titre). Ils sont plus facilement extensibles à tous les types de ressources (utiliser le titre conviendrait pour identifier des peintures, mais si le musée acquiert un fonds de dessins qui n'ont pas de titre particulier, comment les nommer ?). Enfin, ils permettent d'homogénéiser l'identification des ressources, indépendamment de problèmes de langue, d'accentuation, de caractères spéciaux, etc.

Cependant, les identifiants signifiants présentent aussi un certain nombre d'attraits, au premier chef leur facilité d'usage. *Ils sont immédiatement disponibles et ne nécessitent pas la maintenance d'un système de référence distinct*. Enfin, ils permettent de rendre les URIs lisibles en clair pour les humains, ce qui facilite les tâches d'exploitation des données pour les agents humains et parfois pour les agents logiciels (cas de l'optimisation du référencement par les moteurs de recherche).

Dans le cas de l'utilisation d'identifiants signifiants, les règles suivantes devraient être respectées :
- limiter autant que possible l'usage de caractères spéciaux dont le codage peut introduire des ambiguïtés (accents, signes de ponctuation, espaces, etc.) ;
- avoir un usage cohérent de la casse (usage systématique d'une majuscule en début de mot, ou *camel case*, par exemple) ;
- veiller à l'unicité du champ utilisé pour générer l'identifiant à l'intérieur de l'ensemble de données, ou combiner plusieurs champs (par exemple, pour un livre, auteur / titre ; on peut également ajouter dans la syntaxe de l'URI des informations sur la nature de la ressource ou un numéro d'ordre) ;
- maintenir un système capable de lever les ambiguïtés au fur et à mesure qu'elles surviennent (exemple Wikipedia).

D'une façon générale, sauf exception, l'usage d'identifiants signifiants devrait être évité lorsqu'on manipule de très grands ensembles de ressources fortement extensibles car, même si l'on réussit dans un premier temps à circonscrire les problèmes, l'augmentation du nombre de ressources risque de les faire surgir plus tard, et d'obliger à changer de système d'identifiants, ce qui est vraiment la chose la plus pénible pouvant survenir en cours de projet.

2.3 Maintenir et gérer les identifiants dans le temps

Il a longtemps été considéré que le problème de la stabilité des URL (en termes d'unicité mais surtout de pérennité) était un problème technique dont la résolution passait par le choix de *systèmes d'identifiants dits pérennes* (DOI[11], ARK[12], Handle[13] ou INFO[14]).

Ces systèmes d'identifiants pérennes reposent sur l'existence d'une autorité nommante mondialement reconnue, qui attribue à des ressources des identifiants reposant sur la syntaxe des URIs. Pour garantir l'unicité globale de ces URIs, l'enregistrement du *scheme* d'URI est obligatoire auprès de l'IANA[15].

---

[11] Digital object identifier. http://www.doi.org

[12] Archival resource key. https://confluence.ucop.edu/display/Curation/ARK

[13] http://www.handle.net

[14] http://info-uri.info/docs/misc/faq.html

Une première catégorie d'identifiants pérennes a été créée pour pallier les insuffisances des URL telles qu'on les percevait dans les années 1990. Ils ont pour principe d'adopter un autre schème que « http: », puis de définir une organisation pour assigner des autorités nommantes à l'intérieur de ce schème. Le système URN[16] en est un bon exemple : le schème « urn: » est enregistré comme schème d'URI auprès de l'IANA. L'URN contient ensuite un identifiant d'espace de nom (*namespace identifier*) dont l'attribution est gérée par l'IETF[17] : cela correspond au codage d'autorités nommantes comme, par exemple, l'ISBN dans l'identifiant suivant : <urn:isbn:0747591059>.

Ce type de système d'identifiants est purement organisationnel et ne correspond pas à une implémentation technique. Tout comme le schème « info: », créé pour garantir l'unicité globale de systèmes d'identifiants existants qui ne sont pas enregistrés auprès de l'IANA comme schèmes d'URI, ils visent à stabiliser les identifiants dans le temps et non à les rendre actionnables dans le cadre de l'architecture du web.

Au contraire, un système tel que le DOI combine des fonctionnalités organisationnelles et des fonctionnalités techniques. Pour l'organisationnel, il a été créé une entité, l'International DOI Foundation, qui détient la responsabilité de désigner les autorités nommantes et de maintenir les règles de bonnes pratiques de nommage. L'adhésion au système et l'attribution des identifiants sont payantes, mais en retour l'organisation peut garantir leur pérennité au-delà de la durée de vie du producteur.

Du point de vue technique, le DOI utilise une suite logicielle de gestion d'identifiants nommée *Handle*, qui fournit le mécanisme de résolution des identifiants. Grâce à *Handle*, les identifiants sont donc actionnables en passant par un *proxy* ou en installant un *plug-in*. Tout résolveur *Handle* sera capable de résoudre aussi bien des identifiants *Handle* et DOI locaux que ceux d'autres institutions, car le système est globalement cohérent sur le plan technique.

Les règles d'enregistrement de nouveaux schèmes d'URI étant devenues extrêmement restrictives, très peu de systèmes d'identifiants peuvent effectivement être considérés comme des URIs au sens strict du terme. « doi: », « ark: » etc., ne sont pas enregistrés par l'IANA. Ainsi, au sens strict de la norme, l'identifiant <doi:10.1045/july2007-rieger> n'est pas une URI valide, alors que l'identifiant <info:doi:10.1045/july2007-rieger> en est une.

Aujourd'hui, l'idée que le problème n'est pas technique est largement admise, et l'utilisation d'URIs HTTP est recommandée dans la mesure où elles sont les seules à garantir l'intégration des ressources à l'architecture du web. La pérennité est reconnue comme *une problématique organisationnelle, qui repose sur la capacité à imprimer au sein d'une organisation la volonté de s'engager à garantir cette pérennité*, et donc à déployer les processus et les moyens qui s'imposent.

Dans ce contexte, l'adoption d'un système d'identifiants pérennes compatible avec le protocole HTTP (par exemple ARK) peut présenter l'attrait de fournir un cadre au service informatique pour la gestion, et d'obliger l'institution à formuler un engagement de pérennité qui garantit l'affectation des moyens nécessaires. Cela peut aussi aider à harmoniser l'ensemble des identifiants à l'intérieur du système. Toutefois, l'adoption de ces systèmes ne présente en aucun cas une obligation pour garantir la pérennité des URL, et une mauvaise gestion, même si on dispose de DOI ou d'ARK, conduira à l'obsolescence des identifiants tout aussi sûrement que si aucune mesure particulière n'avait été prise pour pérenniser de simples URL.

Le dernier cas à prendre en compte en matière de gestion des identifiants est la disparition des ressources. Il est nécessaire de définir des règles de gestion qui indiquent *ce qu'il advient de l'identifiant au cas où une ressource disparaît* : peut-il ou non être réaffecté à une autre ressource ? Conserve-t-on des informations qui permettent, par exemple, de tracer que la donnée a été détruite ? Au cas où on fusionne deux ressources, qu'advient-il des deux identifiants, restent-ils tous deux valides ? Ici encore il n'existe pas de règle universelle, mais il est nécessaire d'avoir prévu à l'avance ces cas de figure et les comportements qu'ils occasionnent de la part des applications.

3 Encoder et structurer les données

3.1 RDF et le modèle de triplets

Afin de permettre l'échange des données entre les machines, il est nécessaire de disposer d'un formalisme, l'équivalent de la grammaire dans le langage humain, à même d'assurer l'interopérabilité des données sur le web. C'est le but de RDF[18] dont le développement et la normalisation sont assurés par le W3C depuis 1997 [9] [10] [23]. RDF n'est pas un format de fichier, un langage informatique ou un schéma XML, mais un modèle ou un cadre, c'est-à-dire une organisation théorique et logique de l'information.

En RDF, toutes les ressources doivent obligatoirement être identifiées par des URIs. RDF fournit ensuite un formalisme permettant de décrire, sous la forme d'un lien typé, la nature de la relation entre deux ressources désignées par leurs URIs. L'objectif n'est pas la navigation entre ces deux ressources, mais de permettre à des machines d'interpréter la nature de la relation entre elles, alors même qu'elles ne sont pas localisées au même endroit.

Pour ce faire, on compose une assertion dans laquelle le sujet est la première ressource, l'objet la seconde ressource, et le verbe ou prédicat qualifie la nature de leur relation. Cela revient à exprimer toute information sous la forme d'une

---

[15] Internet Assigned Numbers Authority. http://www.iana.org

[16] Uniform resource name

[17] Internet Engineering Task Force

[18] Resource Description Framework

phrase simple, qui va pouvoir être analysée de façon logique par les machines. Toute information doit être exprimée en respectant cette grammaire de base.

Cette assertion, qui suit la structure « sujet – verbe – complément » ou plus précisément « sujet – prédicat – objet », est appelée un triplet.

3.1.1 Le sujet et l'objet

Dans un triplet RDF, le sujet est toujours une ressource désignée par son URI. Dans l'exemple du tableau 1, le sujet du triplet est la personne Eugène Delacroix, représentée par son URI <http://www.mied.fr/personne/Eugene_Delacroix>.

Tableau 1 – Triplets décrivant la personne Eugène Delacroix

Sujet	Prédicat	Objet
<http://www.mied.fr/personne/ Eugene_Delacroix>	<http://www.mied.fr/ontology/ nom>	"Eugène Delacroix"
<http://www.mied.fr/personne/ Eugene_Delacroix>	<http://www.mied.fr/ontology/ prenom>	"Ferdinand Victor Eugène"
<http://www.mied.fr/personne/ Eugene_Delacroix>	<http://www.mied.fr/ontology/ nomFamille>	"Delacroix"
<http://www.mied.fr/personne/ Eugene_Delacroix>	<http://www.mied.fr/ontology/ image>	"http://mied.fr/ 1910_portraits_Eug %C3%A8ne_ Delacroix.jpg"
<http://www.mied.fr/personne/ Eugene_Delacroix>	<http://www.mied.fr/ontology/ nationalite>	"française"
<http://www.mied.fr/personne/ Eugene_Delacroix>	<http://www.mied.fr/ontology/ dateNaissance>	"26 avril 1798"
<http://www.mied.fr/personne/ Eugene_Delacroix>	<http://www.mied.fr/ontology/ lieuNaissance>	"Saint-Maurice"
<http://www.mied.fr/personne/ Eugene_Delacroix>	<http://www.mied.fr/ontology/ dateDeces>	"13 août 1863"
<http://www.mied.fr/personne/ Eugene_Delacroix>	<http://www.mied.fr/ontology/ lieuDeces>	"Paris"
<http://www.mied.fr/personne/ Eugene_Delacroix>	<http://www.mied.fr/ontology/ langueMaternelle>	"Français"
<http://www.mied.fr/personne/ Eugene_Delacroix>	<http://www.mied.fr/ontology/sexe>	"mâle"
<http://www.mied.fr/personne/ Eugene_Delacroix>	<http://www.mied.fr/ontology/ champActivite>	"Peinture"
<http://www.mied.fr/personne/ Eugene_Delacroix>	<http://www.mied.fr/ontology/ champActivite>	"Lithographie"
<http://www.mied.fr/personne/ Eugene_Delacroix>	<http://www.mied.fr/ontology/ voyage>	"Le voyage en Afrique du Nord, au Maroc et retour par l'Espagne et l'Algérie (fin janvier - juillet 1832)"

L'objet du triplet peut être une ressource représentée par une URI ou un littéral (une chaîne de caractère), car toute information n'est pas forcément une entité. Une date, un nombre, un nom... n'est pas une entité ou ressource mais un littéral. Ainsi, lorsque le triplet a pour fonction d'indiquer la relation entre deux ressources, le sujet et l'objet sont tous deux des entités identifiées par des URIs. Quand le triplet a pour fonction de donner une information sur le sujet, de décrire ses caractéristiques, dans ce cas l'objet peut être un littéral.

Il est possible de préciser la nature d'un littéral en attribuant des restrictions à la chaîne de caractère (préciser qu'il s'agit d'un nombre, d'une date, etc.), ce qui permettra de lui appliquer des traitements complémentaires. Dans le tableau 2, les dates de naissances et de décès sont bien qualifiées comme étant des littéraux d'une nature particulière, qui doivent respecter une forme donnée pour être identifiés comme des dates.

Tableau 2 – Triplets décrivant la personne Eugène Delacroix en désambiguïsant les chaînes de caractères

Sujet	Prédicat	Objet
http://www.mied.fr/personne/Eugene_Delacroix	http://www.w3.org/1999/02/22-rdf-syntax-ns#type	http://www.mied.fr/ontologie/Personne
http://www.mied.fr/personne/Eugene_Delacroix	http://www.mied.fr/ontology/nom	"Eugène Delacroix"
http://www.mied.fr/personne/Eugene_Delacroix	http://www.mied.fr/ontology/prenom	"Ferdinand Victor Eugène"
http://www.mied.fr/personne/Eugene_Delacroix	http://www.mied.fr/ontology/nomFamille	"Delacroix"
http://www.mied.fr/personne/Eugene_Delacroix	http://www.mied.fr/ontology/dateNaissance	"1798-04-26"^^xsd:date
http://www.mied.fr/personne/Eugene_Delacroix	http://www.mied.fr/ontology/lieuNaissance	http://www.mied.fr/ville/saint-maurice
http://www.mied.fr/personne/Eugene_Delacroix	http://www.mied.fr/ontology/dateDeces	"1863-08-13"^^xsd:date
http://www.mied.fr/personne/Eugene_Delacroix	http://www.mied.fr/ontology/lieuDeces	http://www.mied.fr/ville/paris
http://www.mied.fr/personne/Eugene_Delacroix	http://www.mied.fr/ontology/estContemporainDe	http://www.mied.fr/personne/Victor_Hugo
http://www.mied.fr/ville/saint-maurice	http://www.w3.org/1999/02/22-rdf-syntax-ns#type	http://www.mied.fr/ontologie/Lieu
http://www.mied.fr/ville/saint-maurice	http://www.mied.fr/ontology/nom	"Saint-Maurice"
http://www.mied.fr/ville/saint-maurice	http://www.mied.fr/ontology/identifiant	"94415"
http://www.mied.fr/ville/saint-maurice	http://www.mied.fr/ontology/parentDivision	http://www.mied.fr/departement/val-de-marne
http://www.mied.fr/departement/val-de-marne	http://www.w3.org/1999/02/22-rdf-syntax-ns#type	http://www.mied.fr/ontologie/Lieu
http://www.mied.fr/departement/val-de-marne	http://www.mied.fr/ontology/nom	"Val-de-Marne"
http://www.mied.fr/departement/val-de-marne	http://www.mied.fr/ontology/identifiant	"94"
http://www.mied.fr/departement/val-de-marne	http://www.mied.fr/ontology/parentDivision	http://www.mied.fr/region/ile-de-france
http://www.mied.fr/ville/paris	http://www.w3.org/1999/02/22-rdf-syntax-ns#type	http://www.mied.fr/ontologie/Lieu
http://www.mied.fr/ville/paris	http://www.mied.fr/ontology/nom	"Paris"@fr
http://www.mied.fr/ville/paris	http://www.mied.fr/ontology/nom	"Parigi"@it
http://www.mied.fr/ville/paris	http://www.mied.fr/ontology/parentDivision	http://www.mied.fr/departement/paris
http://www.mied.fr/departement/paris	http://www.mied.fr/ontology/nom	"Paris"
http://www.mied.fr/departement/paris	http://www.mied.fr/ontology/identifiant	"75"
http://www.mied.fr/departement/paris	http://www.mied.fr/ontology/parentDivision	http://www.mied.fr/region/ile-de-france
http://www.mied.fr/region/ile-de-france	http://www.mied.fr/ontology/nom	"Ile-de-France"

La langue dans laquelle est exprimé le littéral peut être qualifiée par un attribut dont la valeur suit la norme ISO 639-2[19]. La qualification de la langue n'est pas portée par le prédicat, mais par le littéral lui-même.

En l'état, les triplets présentés dans le tableau 1 posent des questions de modélisation : en effet, il est nécessaire d'analyser la nature de chaque information pour déterminer s'il est justifié de la représenter par un littéral (par exemple un nom, un résumé, etc.) ou si elle pourrait être représentée par une URI (par exemple un nom de lieu).

Dans le cas d'un lieu, pour poursuivre sur cet exemple, le réduire à une simple chaîne de caractères (son nom) présente plusieurs désavantages :
- le littéral peut présenter une potentielle ambiguïté du point de vue de l'analyse par les machines. Ainsi, dans le tableau 1, le littéral « Paris » pourrait aussi bien désigner la localité de Paris au Texas que la capitale française ;
- il est impossible de formuler de nouvelles assertions concernant ce lieu.

Dans le tableau 2, grâce à l'utilisation d'une URI pour désigner le lieu « Paris », il devient possible d'associer plusieurs littéraux à une même ressource, ce qui permet par exemple de gérer le multilinguisme. L'entité « Paris » est associée à des appellations en différentes langues (Parigi, etc.).

3.1.2 Le prédicat

En RDF, le prédicat, c'est-à-dire le lien typé qui relie deux ressources, est lui-même une ressource. En tant que tel, il doit donc obligatoirement être identifié par une URI. Il s'agit d'une ressource d'une nature particulière, qu'on appelle une propriété. Nous reviendrons sur cette notion dans la section 3.2.

Il est important de noter que, grâce au modèle de triplet, la nature de la relation entre les deux parties de l'assertion fait partie de la donnée elle-même. Désignée elle aussi par une URI, elle peut elle-même devenir le sujet d'autres assertions.

De ce point de vue, le modèle RDF se différencie des bases de données relationnelles et de XML, dans lesquels le schéma de structuration de l'information est séparé des données. Ici, la référence au système qui permet d'interpréter et d'exploiter la sémantique de la relation est immédiatement disponible, et décrite suivant le même modèle global que la donnée elle-même. Il devient donc possible d'exploiter les données sans connaître *a priori* la sémantique exprimée par le prédicat, et de découvrir cette sémantique en suivant les liens.

3.1.3 Le graphe

Une même ressource peut être sujet, prédicat ou objet dans plusieurs triplets. L'ensemble de ces triplets, reliés les uns aux autres par les URIs qu'ils ont en commun, constitue un graphe. L'intérêt du modèle de graphe réside dans sa souplesse et son évolutivité, puisqu'il est possible d'exprimer une assertion sur une ressource décrite sur un autre serveur sur le web et de manière indépendante aux autres assertions.

Il est important de rappeler que l'organisation logique sous forme de graphe se situe à un niveau complètement abstrait : pour l'instant, nous n'avons pas évoqué le fait d'encoder ces informations à destination des machines.

Le graphe ne constitue pas un ensemble fini d'informations. C'est la somme des triplets qui forme le graphe, et celui-ci n'existe pas en tant que tel. Le graphe existe indépendamment de sa fixation sur un support, qu'il soit physique ou virtuel. Au contraire, par nature, il relie des informations qui peuvent être physiquement stockées sur des serveurs distants. À l'image du graphe de documents constitué par le principe de l'hypertexte, on peut ajouter une donnée ou la retirer sans déstabiliser le graphe dans son ensemble.

Chaque triplet reste une entité indépendante : autoporteur, il doit pouvoir être vérifié indépendamment de quelque contexte que ce soit. La conséquence est que ce triplet ne s'inscrit pas dans un cadre documentaire qui permettrait de garder certaines informations implicites comme ce serait le cas pour des informations stockées à l'intérieur d'une notice. Toute information doit être explicitée.

Par exemple, dans le cas de la notice d'Eugène Delacroix, lorsqu'on mentionne la ville de Paris, l'utilisateur humain devine qu'il s'agit forcément de Paris capitale de la France. Ici c'est le document qui correspond à la notice descriptive de Delacroix qui constitue le contexte implicite permettant à l'utilisateur de compléter l'interprétation d'une information par nature ambiguë. Sur le web sémantique, l'interprétation par les machines ne peut pas s'appuyer sur ce type de contexte ou d'élément implicite ; il faut donc que toute l'information soit exprimée.

3.2 Vocabulaires / ontologies

Si on file la métaphore selon laquelle le web sémantique constitue un langage permettant aux machines de communiquer entre elles, RDF joue le rôle de la grammaire ; il faut ensuite faire appel à des vocabulaires pour qualifier la nature des entités, celle de leurs relations, et leurs caractéristiques. Dans les technologies du web sémantique, ce rôle de « dictionnaire » est assuré par le mécanisme des ontologies exprimées elles-mêmes en RDF à l'aide de deux standards normalisés au sein du W3C : RDFS et OWL.

[19] ISO 639-2:1998 : Codes pour la représentation des noms de langue – Partie 2

3.2.1 Principes de base des ontologies

En philosophie, une ontologie désigne un « discours sur l'être en tant qu'être ». À la suite de Tom Gruber, les chercheurs dans le domaine de l'intelligence artificielle se sont approprié le terme pour désigner une organisation logique et formelle d'un domaine de connaissance pour permettre à une machine d'en manipuler les différents objets, leurs caractéristiques et leurs logiques. Ces ontologies ou vocabulaires permettent de décrire :
- des *classes*, c'est-à-dire des types d'entité du domaine décrit ;
- des *propriétés*, c'est-à-dire les différents types de relations qui unissent entre elles les ressources, et leurs caractéristiques exprimées par des littéraux (chaînes de caractères, dates, nombres, etc.) ;
- une *logique*, c'est à dire la définition de règles ou de comportements associés aux classes et aux propriétés.

Ces classes et ces propriétés se voient attribuer des URIs qui permettent de les désigner sans ambiguïté dans le contexte du graphe global que constitue le web : il devient dès lors possible de partager dans différents contextes une compréhension de ce qu'est « une personne », « un titre », « une œuvre », « une date de naissance »...

3.2.2 Classes et hiérarchies de classes

L'une des fonctions des ontologies est de permettre de définir la nature des ressources. Dans le tableau 2, Eugène Delacroix est de type « personne », Paris est de type « lieu », etc. Ces types d'entité sont appelés des classes. Les classes sont des abstractions auxquelles se rattachent les ressources : on dit alors que les ressources sont des instances de ces classes. Par exemple, la ressource « Eugène Delacroix » est une instance de la classe « personne ».

On désigne la classe par son URI telle qu'elle est déclarée dans l'ontologie. Dans notre exemple, l'URI de la classe « personne » est <http://www.mied.fr/ontologie/Personne>.

L'instance « Eugène Delacroix » est donc reliée à la classe à laquelle elle appartient par une relation dont l'URI est <http://www.w3.org/1999/02/22-rdf-syntax-ns#type>.

Les classes peuvent être organisées hiérarchiquement : on parle alors de classes et de sous-classes. Ce mécanisme permet de déterminer des sous-ensembles spécifiques à l'intérieur d'une même classe : par exemple, une classe « lieu » pourrait avoir comme sous-classe « ville » qui est un type de lieu. Il faut noter que tout ce qui définit une classe peut s'appliquer également aux sous-classes. Ainsi, un « lieu » peut être défini par des coordonnées géographiques ; il en est de même d'une « ville ». En revanche, « ville » n'est pas une sous-classe de « pays » car ces deux entités ne se définissent pas par les mêmes caractéristiques.

En effet, il ne faut pas confondre les rapports que peuvent avoir les instances entre elles avec une hiérarchie de classe. Par exemple, si on dit que « Paris » fait partie de « France », il ne s'agit pas d'une hiérarchie de classe mais d'une relation entre deux instances de la classe « lieu ». « Pays » est un type de lieu et donc une sous-classe de lieu, en revanche « ville » n'est pas un type de pays, donc la relation entre « ville » et « pays » est plutôt une relation de composition (« pays » est composé de « ville ») et non une relation de spécialisation.

On peut également définir dans une ontologie d'autres relations des classes entre elles : par exemple déclarer que deux classes sont disjointes signifie qu'il n'est pas possible de faire partie de l'une et l'autre des deux classes. Par exemple, si on définit une classe « lieu » et une classe « personne », on peut les déclarer comme disjointes et ainsi indiquer qu'une personne ne peut en aucun cas être un lieu.

3.2.3 Les propriétés et leurs caractéristiques

Les propriétés correspondent au prédicat dans le triplet et permettent d'exprimer les relations entre des ressources ou entre une ressource et un littéral. On distingue ainsi les propriétés d'objet (*object property*) qui relient deux ressources entre elles (dans ce cas le sujet et l'objet du triplet sont des URIs), et les propriétés de types de données (*datatype property*) qui relient une ressource à un littéral (le sujet est alors une URI, et l'objet une simple chaîne de caractères).

Dans notre exemple [tableau 2] le prédicat <http://www.mied.fr/ontologie/lieuDeces> est défini comme une propriété d'objet qui permet d'indiquer une relation entre une personne et un lieu, en caractérisant cette relation comme étant le lieu du décès de cette personne.

Le prédicat <http://www.mied.fr/ontologie/nom> est défini comme une propriété de type de données qui permet d'indiquer le nom d'une personne sous la forme d'un littéral.

L'ontologie permet d'indiquer le comportement d'une propriété par rapport aux ressources qu'elle qualifie : une propriété se définit par la classe à laquelle appartiennent les ressources qui seront sujet et objet d'un triplet dont la propriété est le prédicat.

Le domaine (*domain* en anglais) de la propriété détermine la classe des ressources qui peuvent être sujet d'une propriété, et son codomaine (*range* en anglais) détermine soit la classe des ressources qui peuvent être l'objet du triplet s'il s'agit d'entités, soit le type de données dans le cas d'un littéral.

Ainsi, dans notre ontologie, la propriété <http://www.mied.fr/ontologie/lieuDeces> a pour domaine <http://www.mied.fr/ontologie/Personne> et pour codomaine <http://www.mied.fr/ontologie/Lieu>. Cela signifie que les triplets qui ont pour prédicat cette propriété ont forcément pour sujet une ressource de la classe « personne » et pour objet une ressource de la classe « lieu ».

La propriété <http://www.mied.fr/ontologie/dateNaissance> est définie dans l'ontologie comme ayant pour codomaine une chaîne de caractères d'une forme particulière qui correspond à une date. Pour cela, on fait référence à des types de données définis dans la norme *XML Schema* [14] et identifiés par une URI, dans ce cas <http://www.w3.org/2001/XMLSchema#date>.

On peut aussi utiliser l'ontologie pour préciser la logique associée aux propriétés, c'est-à-dire des règles et des comportements qui les définissent ou définissent les relations qu'elles peuvent avoir entre elles.

Il est par exemple possible de déterminer qu'une propriété est symétrique, ce qui signifie qu'elle s'applique de manière réciproque entre deux ressources : si Eugène Delacroix est contemporain de Victor Hugo (propriété <http://www.mied.fr/ontologie/estContemporainDe>) et que la propriété <http://www.mied.fr/ontologie/estContemporainDe> est déclarée symétrique, cela signifie que Victor Hugo est également contemporain d'Eugène Delacroix.

Nous reviendrons plus loin sur les différents types de comportements qu'il est possible de définir dans une ontologie.

Enfin, comme les classes, les propriétés peuvent être organisées hiérarchiquement, les sous-propriétés héritant des caractéristiques de la propriété qu'elles spécialisent, notamment de son ou ses comportements, de son domaine, et de son codomaine.

L'intérêt réside dans la possibilité de définir des propriétés plus précises : par exemple, on pourrait avoir une propriété générique définissant la relation entre une personne et un lieu, qui serait précisée par les sous-propriétés <http://www.mied.fr/ontologie/lieuNaissance> et <http://www.mied.fr/ontologie/lieuDeces>. Cette propriété générique ayant pour codomaine la classe « lieu », alors il en est de même pour les deux sous-propriétés. On pourrait aussi leur affecter un codomaine plus précis, mais seulement à condition que la classe de celui-ci soit elle-même une sous-classe de celle qui est indiquée en codomaine de la propriété plus générique.

Enfin, si une sous-propriété s'applique entre deux ressources, alors la propriété plus générique doit elle aussi pouvoir s'appliquer.

3.2.4 RDFS et OWL

Le W3C a mis au point deux recommandations pour décrire des ontologies ou vocabulaires en RDF : RDFS et OWL.

Selon la définition de la recommandation [11], RDFS[20], chronologiquement antérieur à OWL, est un langage de description de vocabulaires RDF dont l'objectif est de déclarer des classes et des propriétés de manière très simple.

RDFS permet de déclarer la logique hiérarchique des classes et propriétés, c'est-à-dire l'arborescence des classes et sous-classes, propriétés et sous-propriétés, et de définir les domaines et codomaines des propriétés. C'est également dans le standard RDFS que sont définies les modalités d'instanciation d'une classe : le mécanisme qui permet de rattacher une instance à sa classe à l'aide de la propriété <http://www.w3.org/1999/02/22-rdf-syntax-ns#type>.

L'avantage de RDFS réside dans sa simplicité, qui en fait un standard facile à manipuler, mais qui n'a pas été conçu pour exprimer une logique complexe. Il n'offre qu'une expressivité limitée quant au type de contraintes que l'on peut utiliser pour caractériser les classes et les propriétés.

OWL[21] introduit des éléments qui permettent de contraindre et donc de mieux préciser la logique formelle d'un vocabulaire [13] [20]. Nous en citons ici quelques exemples de façon non exhaustive.

Contraintes sur les classes :
- possibilité de déclarer le fait qu'une classe est composée d'une liste d'instances particulières, ce qui permet de déterminer une classe en la limitant à une liste contrôlée ;
- possibilité de définir que toute ressource appartenant à une classe est caractérisée par une propriété donnée (par exemple la classe « personne » a forcément une propriété « date de naissance ») ;
- définition de relations autres que hiérarchiques entre les classes, par exemple le fait qu'une classe est l'union ou l'intersection de deux classes, ou le fait que deux classes sont disjointes ou complémentaires.

Contraintes sur les propriétés :
- propriétés symétriques (si A est frère de B, alors B est frère de A) ;
- propriétés inverses (si A est parent de B, alors B est enfant de A) ;
- propriétés transitives (si A est frère de B et B est frère de C, alors A est frère de C) ;
- cardinalité des propriétés (A peut avoir de 0 à N frères ; A a exactement 1 père).

L'une des fonctions les plus intéressantes de OWL est la déclaration d'équivalence, sur laquelle nous allons nous attarder un peu ici car elle joue un rôle tout particulier dans le web sémantique et dans le web de données. Il est en effet possible de déclarer comme équivalentes :
- des classes : <http://www.w3.org/2002/07/owl#equivalentClass> ;
- des propriétés : <http://www.w3.org/2002/07/owl#equivalentProperty> ;
- des instances : <http://www.w3.org/2002/07/owl#sameAs>.

L'utilité de la déclaration d'équivalence est de permettre de réconcilier des ontologies qui auraient été déclarées séparément mais traiteraient des mêmes objets. Par exemple, nous avons déclaré la classe <http://www.mied.fr/ontologie/Personne>. Imaginons que le musée Victor Hugo déclare également une classe ayant pour URI <http://www.museevictorhugo.fr/ontologie/Personne>. Alors nous pourrions créer une assertion reliant ces deux classes par la propriété « equivalentClass », ce qui permettrait à une machine de comprendre que ces deux classes sont interchangeables et que l'entité <http://www.mied.fr/personne/Eugene_Delacroix> appartient en fait à la même classe que l'entité <http://www.museevictorhugo.fr/Victor_Hugo>.

[20] RDF Schema

[21] Web Ontology Language

De la même manière, pour les instances, la propriété <owl:sameAs> va permettre à des jeux de données relevant d'institutions différentes d'indiquer qu'ils évoquent des objets similaires. On pourrait par exemple déclarer :

<http://www.mied.fr/personne/Eugene_Delacroix>
<http://www.w3.org/2002/07/owl#sameAs>
<http://www.museevictorhugo.fr/Eugene_Delacroix>

Ceci est possible à condition qu'il s'agisse bien des mêmes entités, et que toutes les assertions concernant <http://www.mied.fr/personne/Eugene_Delacroix> puissent également s'appliquer sans créer d'incohérence à l'URI <http://www.museevictorhugo.fr/Eugene_Delacroix>. En effet, l'implication logique de l'équivalence entre les instances est extrêmement forte puisqu'elle implique une identité complète entre les deux instances reliées : de façon sommaire, on indique que deux URIs désignent en fait exactement la même entité abstraite. Dès lors, toutes les assertions formulées concernant l'une des deux ressources s'appliquent aussi à l'autre.

Ce système évite de se retrouver dans une situation complexe où toutes les institutions du monde devraient se mettre d'accord sur le nommage des entités abstraites avant de pouvoir commencer à formuler la moindre assertion. Nous avons vu dans la section 2 la complexité que représente l'attribution des URIs aux entités non documentaires. Si un tel système devait être défini *a priori*, ce serait un facteur d'échec presque certain pour le web sémantique. Grâce à la possibilité de déclarer des instances comme équivalentes, on dispose d'un système de liens relativement souple qui autorise une même entité à recevoir plusieurs URIs distinctes sans que cela ne créée d'incohérence dans le graphe global.

3.2.5 Le principe des inférences

Une ontologie, décrite en OWL ou en RDFS, ne permet pas à proprement parler de valider les données encodées en RDF comme c'est le cas par exemple avec un schéma XML. En effet, en RDF, la logique structurelle est toujours la même, puisqu'elle est intrinsèque au modèle « sujet- prédicat-objet ». *RDF ne s'intéresse donc pas à l'encodage d'une structure, mais plutôt à celui de la logique des données*.

Le but de l'ontologie est de définir le cadre logique et sémantique d'un domaine de connaissances, en utilisant les contraintes appliquées aux classes et propriétés comme nous l'avons décrit ci-dessus. Une fois l'ontologie déclarée de manière formelle avec un langage comme OWL ou RDFS, l'ensemble des assertions disponibles peut être analysé automatiquement par un logiciel qu'on appelle raisonneur. Celui-ci combine l'information disponible, c'est-à-dire les contraintes exprimées dans l'ontologie et les assertions exprimées dans les données, pour l'analyser et en déduire de nouvelles informations. On appelle ce mécanisme l'inférence. L'application des mécanismes d'inférence permet de créer de nouvelles assertions de manière logique à partir de celles qui sont déjà exprimées.

L'analyse conduite par la machine n'est pas une validation, au sens où elle ne va pas chercher à vérifier que toutes les contraintes déclarées dans l'ontologie sont bien respectées dans les données. Au contraire, si une information n'est pas présente, elle va chercher à la compléter par inférence.

Par exemple, on peut inférer le type d'une ressource à partir de l'application d'une propriété, en fonction de ses domaines et codomaines. Nous avons vu plus haut que la propriété <mied:lieuDeces> dans notre ontologie a pour domaine <mied:Personne> et pour codomaine <mied:Lieu>. Ainsi, le triplet :

<http://www.mied.fr/personne/Eugene_Delacroix>
<http://www.mied.fr/ontologie/lieudeces>
<http://www.mied.fr/ville/paris>

permet à une machine de déduire que <http://www.mied.fr/personne/Eugene_Delacroix> est de type <http://www.mied.fr/ontologie/Personne>, et que <http://www.mied.fr/ville/paris> est de type <http://www.mied.fr/ontologie/Lieu>.

Autre exemple : il est possible de raisonner sur les comportements de propriétés pour créer de nouveaux triplets. Nous avons vu plus haut que la propriété <mied:estContemporainDe> est déclarée symétrique dans notre ontologie. Le triplet :

<http://www.mied.fr/personne/Eugene_Delacroix>
<http://www.mied.fr/ontologie/estContemporainDe>
<http://www.mied.fr/personne/Victor_Hugo>

permet dès lors d'inférer le triplet :

<http://www.mied.fr/personne/Victor_Hugo>
<http://www.mied.fr/ontologie/estContemporainDe>
<http://www.mied.fr/personne/Eugene_Delacroix>

Le principe des inférences va être utilisé lorsqu'on exploitera les données pour tirer parti de la souplesse et de la richesse du modèle. Ce principe ne se limite pas à l'exploitation de la logique inhérente aux ontologies. Il est également possible de définir des règles qui permettent d'exploiter la sémantique des données dans un contexte spécifique : des règles « métier » qui indiquent des modalités d'extraction de nouvelles informations à partir d'un graphe existant. C'est ce que nous allons voir à présent avec SPARQL.

3.3 Le protocole et langage de requête SPARQL

Afin d'interroger un ensemble de données structurées en RDF, le W3C a mis au point un langage de requêtes dédié, SPARQL [17] [18] [19] [22]. En fournissant un formalisme pour explorer le contenu d'un graphe RDF, SPARQL joue un rôle équivalent à celui du langage de requête SQL dans le monde des bases de données relationnelles. Il est ainsi possible d'interroger les données dans toute leur richesse et d'exploiter pleinement le modèle de graphe.

3.3.1 Principes de base de SPARQL

SPARQL[22] offre les moyens de parcourir un graphe RDF selon un principe simple qui pourrait être comparé à celui des équations en mathématiques. Une requête SPARQL a pour but de trouver un ou plusieurs membres d'un ensemble de triplets en posant une ou plusieurs inconnues, ou variables, qui sont le sujet, le prédicat, et/ou l'objet d'un triplet. Ainsi une requête formulée de la manière suivante :

```
?sujet ?prédicat ?objet
```

permet de sélectionner l'ensemble des triplets du graphe qu'on est en train d'interroger, quel que soit leur sujet, leur prédicat ou leur objet. On va ensuite remplacer certaines variables par des URIs ou des littéraux de façon à ramener un sous-ensemble de résultats. Par exemple :

```
<http://www.mied.fr/personne/Eugene_Delacroix> ?prédicat ?objet
```

permet de sélectionner tous les triplets dont la ressource Eugène Delacroix est le sujet.

Voici quelques exemples de requêtes SPARQL simples qu'on pourrait effectuer sur notre exemple :
- retrouver un littéral en objet : Quelle est la date de naissance d'Eugène Delacroix ?
- retrouver une ressource en objet : Quel est le lieu de naissance d'Eugène Delacroix ?
- retrouver un sujet : Quelles sont les œuvres dont Eugène Delacroix est le créateur ?

On peut ensuite combiner plusieurs de ces équations entre elles afin de préciser la requête, ce qui revient à parcourir le graphe. SPARQL permet ainsi de formuler des inférences simples en utilisant les contraintes exprimées dans l'ontologie et les notions de classes et de propriétés. Exemples :
- interroger les ontologies (classes) : Quels sont les types d'œuvres liés à Eugène Delacroix ?
- retrouver les prédicats et interroger les ontologies (propriétés) : Quels sont les rôles joués par Eugène Delacroix dans les œuvres auxquelles il est lié ?

L'intérêt du parcours de graphe en SPARQL réside aussi dans la possibilité d'enchaîner des triplets dont certains éléments ne sont pas connus. On peut ainsi découvrir des éléments qui sont éloignés de l'objet de notre recherche par des inconnues. Exemples :
- traverser le graphe sans connaître l'objet 1 : Quelles sont les personnes décédées dans la même ville qu'Eugène Delacroix ?
- traverser le graphe sans connaître l'objet 2 : Quels sont les œuvres des personnes influencées par Eugène Delacroix ?
- traverser le graphe sans connaître le sujet 1 : Dans quel musée sont exposées actuellement les œuvres d'Eugène Delacroix ?
- traverser le graphe à deux niveaux : Quels sont les auteurs de livres sur Eugène Delacroix ?
- traverser le graphe à trois niveaux : Dans quelles villes se situent les peintures d'Eugène Delacroix ?

Outre ces fonctions de parcours de graphe, SPARQL fournit également des fonctions avancées (négations, options, filtres, troncatures, comptages, etc.). Exemples :
- fonction d'agrégation : Combien de livres ont pour thème Eugène Delacroix ?
- agrégation + sous-requêtes : Dans quelle ville trouve-t-on le plus de peintures d'Eugène Delacroix ?
- fonctions de négation : Quels sont les peintres qui ne sont pas romantiques ?

Enfin, une fonction de construction de graphe permet de générer un nouveau graphe à partir du graphe interrogé, éventuellement en y intégrant des inférences : c'est-à-dire qu'on peut produire en SPARQL des triplets RDF qui n'existaient pas dans le graphe initial. Exemple : si Eugène Delacroix est romantique, alors ses œuvres relèvent du courant romantique.

[22] SPARQL Protocol And RDF Query Language

3.3.2 Le protocole et le flux XML de réponses

SPARQL comporte une dimension supplémentaire : le W3C ne s'est pas limité à définir le langage de requête pour interroger le graphe, mais il a aussi élaboré le protocole [18] permettant d'envoyer la requête sur le web et les formalismes de réponse sous la forme d'un flux XML [19].

Cela permet à SPARQL de jouer également le rôle d'interface de programmation entre deux applications (API[23]), à ceci près que la sémantique de l'interrogation se situe dans le graphe lui-même et non dans l'API : il devient possible dès lors d'utiliser le même langage de requête pour n'importe quel graphe de données exprimé en RDF, indépendamment des spécificités de l'application. Ici encore c'est la puissance de la normalisation et de l'architecture du web qui permet de s'approcher de l'idée d'API universelle.

SPARQL fournit également le format et la structure du flux XML de réponses, de façon à permettre à la machine qui effectue la requête d'exploiter de manière standard les résultats obtenus.

Il est à signaler que le W3C travaille actuellement à une nouvelle version des différentes recommandations liées à SPARQL qui introduira, entre autres, les fonctionnalités de mises à jour de données RDF, d'interrogations de plusieurs ensembles de données dans une même requête, d'agrégation (calcul de somme, de moyenne...) et développera les API.

3.4 La sérialisation

Comme nous l'avons vu, RDF définit un modèle logique, abstrait, pour décrire la sémantique des informations, jouant le rôle d'une grammaire, tandis que les ontologies fournissent le vocabulaire dans lequel on va puiser pour exprimer ces informations de façon normée.

Pour inscrire le langage naturel sur un support en vue de son partage de façon asynchrone, nous disposons d'un système graphique d'expression du langage : l'écriture. De même, pour que les machines puissent interpréter les informations exprimées suivant le modèle RDF, il est nécessaire d'exprimer les données suivant une syntaxe spécifique. Ainsi, tout en respectant les principes de RDF, les assertions peuvent être sérialisées selon plusieurs syntaxes : RDF/XML, N3, N-triples, Turtle, RDFa, etc.

3.4.1 Le principe des espaces de noms et des préfixes

Par convention, on abrège les noms des classes et des propriétés à l'aide d'un préfixe qui correspond à l'ontologie dans laquelle elles sont déclarées. Dans notre exemple, le préfixe « mied » correspond à l'URI <http://www.mied.fr/ontologie/>, ce qui signifie qu'écrire <mied:Personne> revient à écrire l'URI suivante : <http://www.mied.fr/ontologie/Personne>.

Quelle que soit la syntaxe de sérialisation, il est d'usage de déclarer les préfixes utilisés au début du « document », suivant un formalisme propre à chaque syntaxe.

3.4.2 Les syntaxes N3, N-triples, Turtle

Non encore normalisées, ces différentes syntaxes reposent, à quelques différences minimes près, sur un principe très simple qui s'apparente à des phrases se terminant par un point. La syntaxe Turtle est en cours de normalisation au W3C [21].

Cette sérialisation repose sur un fichier texte comprenant en premier lieu la déclaration des préfixes des espaces de noms utilisés dans les triplets qui suivent, puis l'ensemble des triplets. Chaque triplet se termine par un point. Si deux triplets qui se suivent ont le même sujet, le sujet n'est pas répété et un point-virgule sépare l'objet du premier et le prédicat du second. Si deux triplets qui se suivent ont le même sujet et le même prédicat, le sujet et le prédicat ne sont pas répétés et une virgule sépare les différents objets. Exemple :

```
@prefix rdfs: <http://www.w3.org/2000/01/rdf-schema#> .
@prefix rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#> .
@prefix mied: <http://mied.fr/ontologie/> .
<http://www.mied.fr/personne/Eugene_Delacroix> rdf:type mied:Personne ;
 mied:nom "Eugène Delacroix".
```

3.4.3 La syntaxe RDF/XML

Si les triplets peuvent être écrits selon les principes de la syntaxe XML, il ne faut pas pour autant en conclure que RDF est un schéma de XML et confondre les deux. En effet, XML est à la fois une syntaxe et un modèle qui repose sur l'idée d'organisation hiérarchique de l'information. Dans le cadre du RDF/XML [12], c'est la syntaxe uniquement qui est utilisée. De fait, il n'est pas possible de valider un fichier RDF/XML par rapport à un schéma comme les autres fichiers utilisant XML. En outre, le système des espaces de noms n'est pas utilisé dans le même sens.

L'élément racine est toujours RDF. Le système des espaces de noms qui désignent en XML un élément rattaché à un schéma XML constitue pour RDF un mécanisme d'abréviations des URIs, comme nous l'avons vu en 3.4.1.

[23] Application programming interface

```
<rdf:RDF
 xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
 xmlns:rdfs="http://www.w3.org/2000/01/rdf-schema#"
 xmlns:mied="http://mied.fr/ontologie/">
</rdf:RDF>
```

Une ressource décrite est toujours introduite par l'élément <rdf:Description>, l'URI de la ressource (le sujet) se trouvant dans l'attribut @rdf:about. Les éléments enfants sont les prédicats qui se rattachent à cette même ressource. Si l'objet est un littéral, il est représenté sous la forme de la valeur d'un élément. Si l'objet est une ressource, son URI est indiqué dans l'attribut @rdf:resource.

```
<?xml version="1.0" encoding="UTF-8"?>
<rdf:RDF
xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
 xmlns:rdfs="http://www.w3.org/2000/01/rdf-schema#"
 xmlns:mied="http://mied.fr/ontologie/">

<rdf:Description rdf:about="http://www.mied.fr/personne/Eugene_Delacroix">
        <rdf:type rdf:resource="http://mied.fr/ontologie/Personne"/>
        <mied:nom>Eugène Delacroix</mied:nom>
 </rdf:Description>
</rdf:RDF>
```

Dans la mesure où RDF/XML est une représentation d'un graphe sous la forme d'un arbre, il existe plusieurs manières de représenter les mêmes triplets. Ainsi, l'exemple précédent pourrait être sérialisé sans aucune perte d'informations de la manière suivante :

```
<?xml version="1.0" encoding="UTF-8"?>
<rdf:RDF
        xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
        xmlns:rdfs="http://www.w3.org/2000/01/rdf-schema#"
        xmlns:mied="http://mied.fr/ontologie/">

 <mied:Personne rdf:about=" http://www.mied.fr/personne/Eugene_Delacroix " mied:nom="Eugène
Delacroix">
 </mied:Personne>
</rdf:RDF>
```

Dans ce cas, le type de la ressource prend la place de l'élément rdf:Description, l'attribut de cet élément correspond à un prédicat et sa valeur à l'objet.

3.4.4 RDFa

HTML, de même que XHTML, permet de structurer une page web selon les principes d'un langage à balises. Les différentes balises indiquent de manière hiérarchique le rôle joué par chaque portion d'information dans le contexte de la page web. Pourtant, pour exploiter automatiquement l'information, il pourrait être utile d'exprimer dans le code HTML la structure ou la description du message quel que soit le code HTML. C'est précisément le but de RDFa[24] [24] qui s'appuie sur le modèle RDF.

Pour ce faire, RDFa précise l'utilisation d'attributs existants en HTML et en introduit de nouveaux. Il devient ensuite possible, à partir de ces différents attributs et de la structure hiérarchique des éléments HTML, d'extraire les triplets de la page web pour analyser ou traiter les informations.

L'introduction dans une page web des triplets exprimés précédemment pourrait donner le code HTML suivant :

```
<html xmlns="http://www.w3.org/1999/xhtml"
        version="XHTML+RDFa 1.1"
        xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
        xsi:schemaLocation="http://www.w3.org/1999/xhtml
http://www.w3.org/MarkUp/SCHEMA/xhtml-rdfa-2.xsd"
prefix="mied: http://mied.fr/ontologie/">
 <head>
        <title>Eugène Delacroix au Musée Imaginaire</title>
```

[24] Resource Description Framework in Attributes

```
</head>
<body>
        <div typeof="mied:Personne"
        about="http://www.mied.fr/personne/Eugene_Delacroix">
        Bienvenue sur le site du Musée Imaginaire
        <span property="mied:nom">Eugène Delacroix</span>.
        </div>

</body>
</html>
```

L'URI du sujet est indiquée avec l'attribut @about, son type avec l'attribut @typeof, les prédicats sont indiqués par l'attribut @property, l'objet étant la valeur de l'élément qui contient cet attribut. Dans le cas où l'objet est une ressource, l'attribut est @rel et l'objet est indiqué dans un attribut @href.

4 Relier, réutiliser, partager : l'apport du web de données

L'expression *web of data*, qu'on traduit de manière littérale par « web de données », est présente dès la feuille de route pour le web sémantique écrite en 1998 par Tim Berners-Lee [3]. Cependant, elle n'a été vraiment utilisée qu'à partir de 2006, suite à la parution de la note *Linked Data* du même Tim Berners-Lee [4]. Faisant face à un relatif constat d'échec dressé par la communauté, cette note visait à rappeler les buts initiaux poursuivis par le web sémantique, dans un contexte où les technologies développées par le W3C étaient surtout utilisées dans le cadre d'expérimentations limitées au domaine de la recherche, et peinaient à voir se développer une réelle adoption dans des applications industrielles. Les laboratoires de recherche qui travaillaient à produire des données en RDF ou des ontologies se préoccupaient trop peu de la publication de ces données sur le web, et l'énergie consacrée à la modélisation de vastes domaines de connaissance finissait par nuire à la réutilisation des données, qui requiert une forme de simplicité.

La note de 2006 rappelle le but initial du web sémantique, à savoir établir des liens entre les données exposées et distribuées sur le réseau, et elle contient les quatre principes de mise à disposition des données grâce aux technologies du web sémantique. Elle a constitué le point de départ d'une renaissance avec le projet *Linking Open Data* visant à offrir des cas d'utilisation réels et simples. L'hypothèse posée par les membres de ce groupe était qu'il était nécessaire de disposer de premiers jeux de données (*datasets*) en RDF, sous licence libre, afin de constituer un point de départ pour démontrer l'intérêt de la démarche et permettre à de nouveaux jeux de données de venir s'y relier. La mise à disposition de DBpedia, qui constitue une extraction en RDF des données de l'encyclopédie collaborative Wikipédia, a constitué un tournant dans l'histoire du web de données. Elle a fourni un ensemble de données en RDF suffisamment massif et généraliste pour susciter une grande variété d'usages et permettre à différents acteurs de trouver un point d'ancrage pour se relier à d'autres données.

En novembre 2009, on pouvait encore tenter de dénombrer les triplets disponibles sur le web de données : il était alors constitué de 13,1 milliards de triplets répartis au sein de différents ensembles de données couvrant les domaines aussi diverses que les données multimédia, les données du web social, les données géographiques et statistiques, les données bibliographiques...

4.1 Une interopérabilité basée sur les liens

Le web de données propose une forme d'interopérabilité qui ne repose ni sur l'interrogation synchrone de bases réparties (comme le protocole Z3950 ou la plupart des *services web* utilisés pour créer des portails), ni sur la réduction de bases diverses à un format commun (comme dans le protocole OAI-PMH combiné avec l'utilisation du Dublin Core simple), mais sur la création d'un espace global d'information, utilisant les liens pour permettre de naviguer de manière transparente d'une ressource à l'autre.

Les règles de bonnes pratiques du web de données, énoncées par Tim Berners-Lee dans la note de 2006 puis adaptées par le groupe SWEO[25] dans le cadre de l'initiative *Linking Open Data*, sont au nombre de quatre :
- utiliser des URIs pour identifier les ressources : chaque ressource sur laquelle on veut pouvoir faire des assertions doit se voir affecter une URI ;
- ces URIs doivent être formulées suivant le protocole HTTP afin qu'on puisse les actionner pour accéder à la ressource identifiée ou à des informations sur cette ressource ;
- lorsqu'on accède à une ressource via son URI, celle-ci doit renvoyer des informations utiles et pertinentes en utilisant les standards (RDF, SPARQL) ;
- enfin, les ressources doivent être reliées, c'est-à-dire qu'il ne suffit pas de publier des informations, mais il faut les relier à des informations publiées par d'autres, afin de créer un écosystème basé sur les liens.

Ces règles montrent bien que le but du web de données n'est pas de créer un autre web, puisqu'il s'appuie sur son architecture actuelle (le système des URIs et le protocole HTTP), mais d'en créer une extension. Ainsi, RDF est aux

[25] Semantic Web Education and Outreach

données structurées ce que HTML est aux documents : un cadre d'interopérabilité qui permet d'assurer une cohérence dans la manipulation et le traitement de ces données par les machines. L'objectif est de créer un espace global d'information où les données sont décrites suivant un modèle commun, le modèle RDF, et reliées par des liens actifs, exploitables par des machines.

Grâce aux principes du modèle RDF, les liens entre les données sont typés, c'est-à-dire qu'ils qualifient la nature de la relation qui relie deux ressources : similarité, relation de sujet (*aboutness*), ou autre. Dans cette approche, il est possible de créer des liens entre des ressources décrites en utilisant divers modèles, à partir du moment où la grammaire de base, commune à tous ces modèles, est le RDF.

Deux modèles d'interopérabilité permettent de représenter cette nouvelle façon de travailler les données : le modèle de la roue et de l'essieu (*hub and spoke*) et le modèle de la navigation intuitive (*follow your nose*).

Les référentiels ou vocabulaires sont appelés à jouer un rôle vital dans le web de données, en particulier lorsqu'il s'agit de construire l'interopérabilité entre des données issues de domaines différents. Sur le web, un utilisateur a la possibilité de naviguer d'un site à un autre sans avoir connaissance des moyens techniques utilisés pour publier les données, sans même qu'il n'existe véritablement de rupture ou de frontière entre ce qu'on appelle les sites web. De la même manière, sur le web de données, la navigation de lien en lien doit pouvoir se faire, d'un jeu de données à un autre, sans nécessité de percevoir les limites des différentes bases de données ni leur format.

Les référentiels tels que thésaurus, vocabulaires contrôlés, listes d'autorité, etc. [voir le chapitre 4], sont volontiers associés au modèle *hub and spoke* : ils agissent comme un point nodal ou une colonne vertébrale permettant de créer un point de contact entre des jeux de données différents [figure 3]. Dans le web de données, ce point de contact est suffisant pour naviguer sans contrainte d'un ensemble à l'autre, en utilisant les URIs, que les données soient ou non exprimées suivant le même modèle.

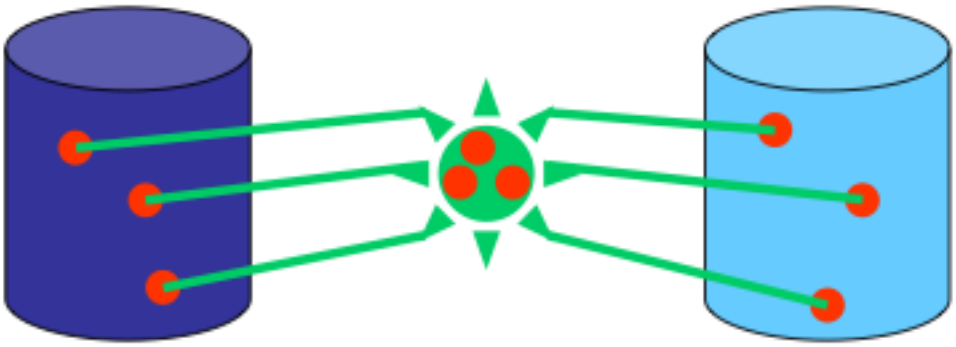

Figure 3 – Interopérabilité basée sur les liens : modèle *hub and spoke*

Pour aller encore plus loin, dans le web de données, n'importe quel jeu de données dont on réutilise les entités peut jouer ce même rôle de passerelle, quoique pas de manière centralisée : le fait de parcourir ces liens permet alors de découvrir de nouvelles ressources de façon intuitive (*follow your nose interoperability*), simplement en naviguant d'un jeu de données à un autre au gré des liens [figure 4].

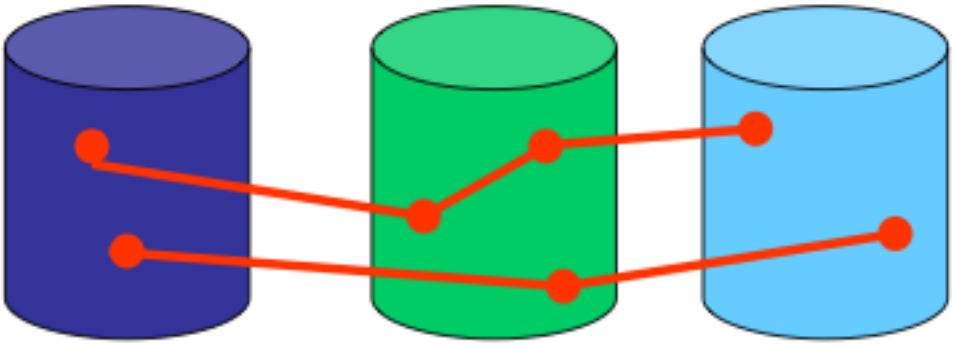

Figure 4 – Interopérabilité basée sur les liens : modèle *follow your nose*

4.2 La négociation de contenu

L'interopérabilité par les liens, ou modèle *follow your nose*, impose que lorsqu'une URI est actionnée, ou « déréférencée » par une machine ou un utilisateur, ceux-ci doivent obtenir en retour de l'information utile, c'est-à-dire qu'ils soient capables de traiter. Concrètement, cela signifie qu'une URI exposée selon les principes du web de données doit renvoyer une représentation adaptée au type de client qui a effectué la requête : une page HTML s'il s'agit d'un utilisateur humain via un navigateur web, et un flux encodé suivant une des sérialisations de RDF si la requête a été effectuée par une machine qui a déclaré explicitement préférer ce type de format. Ce processus s'appelle la négociation de contenu [figure 5].

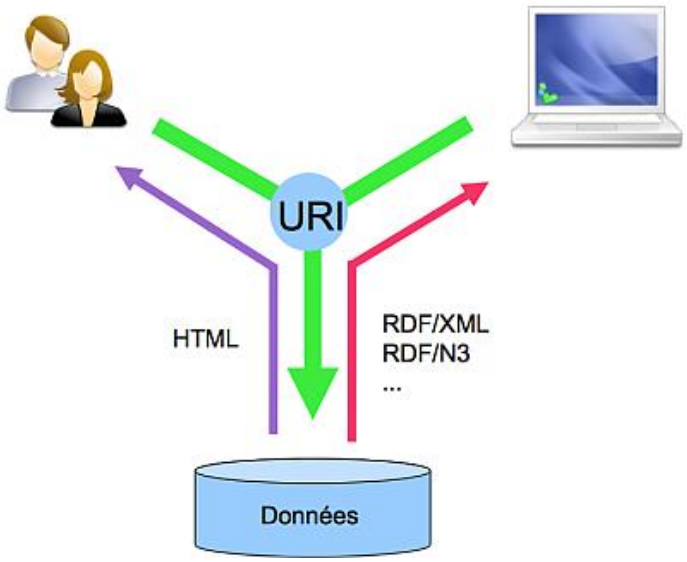

Figure 5 – Le processus de « négociation de contenu »

Pour pouvoir procéder à la négociation de contenus, il est nécessaire d'affecter au moins trois URIs à chaque ressource :
- une URI abstraite correspondant à l'identifiant de la ressource, par exemple <http://www.mied.fr/personne/Eugene_Delacroix> ;
- une URI correspondant à la représentation HTML de la ressource, par exemple <http://www.mied.fr/personne/Eugene_Delacroix.html> ;
- une URI correspondant à la représentation en RDF/XML de la ressource, par exemple <http://www.mied.fr/personne/Eugene_Delacroix.rdf>.

En fonction des critères de la requête HTTP sur la ressource abstraite, <http://www.mied.fr/personne/Eugene_Delacroix>, le serveur va rediriger le client vers une des deux autres ressources. Dans le cas où la requête est issue d'un navigateur web, le serveur redirigera vers la représentation en HTML, et dans le cas où la requête est issue d'un programme informatique qui sait interpréter le RDF/XML, le serveur redirigera vers la représentation en RDF/XML. Si les informations ne permettent pas au serveur d'effectuer cette négociation, il renverra une des représentations par défaut, à savoir celle qui est paramétrée au niveau du serveur.

Chaque représentation de la ressource regroupe en fait une portion du graphe, un ensemble de triplets qui sont sélectionnés parce qu'ils sont considérés comme significatifs pour représenter la ressource. Par exemple, il est d'usage que la représentation RDF/XML regroupe tous les triplets dont la ressource est sujet ou objet. Le contenu de la page HTML peut rassembler davantage d'information, comme les labels associés aux ressources liées afin de les rendre plus explicites pour l'utilisateur.

4.3 Relier les différents ensembles de données

La création des liens, qui constituent une brique essentielle du web de données et l'application du quatrième des principes édictés par Tim Berners-Lee, s'avère complexe lorsque les données sont issues de la conversion de silos de données existants qui, par nature, ne sont pas reliés à d'éventuels autres ensembles de données. Des mécanismes doivent être utilisés pour transformer des systèmes de références non globaux (par exemple des clés internes de bases de données) en URIs, pour exploiter des systèmes d'identifiants externes à l'architecture du web (par exemple les ISBN, ISSN, etc.), ou encore pour créer automatiquement des liens à partir de l'analyse de chaînes de caractères. Ces automatismes s'appliquent avec plus ou moins de succès suivant la nature des jeux de données qu'on tente d'aligner, leur homogénéité ou au contraire leur diversité, et leur niveau de normalisation [voir le chapitre 4].

De fait, la nature des liens qui ont pu être créés jusqu'à maintenant dans la plupart des cas est souvent assez pauvre, se limitant dans une majorité des cas à indiquer une équivalence d'identité entre deux ressources avec la propriété <owl:sameAs>. Pour pallier cet état de fait et éviter qu'il ne constitue un blocage à la publication des jeux de données,

les tenants du mouvement de l'*open data* ont travaillé en faveur de l'assouplissement des règles, remplacées par un système de gradation croissante qui va de la mise à disposition des données sur le web, quel qu'en soit le format, au respect des quatre principes du web de données.

Malgré tout, au sein des 13 milliards de triplets que représentait le web de données en novembre 2009, on comptait 142 millions de liens entre les ensembles de données qu'on a pris l'habitude de représenter sous la forme dite *Linking Open Data Cloud* [figure 6].

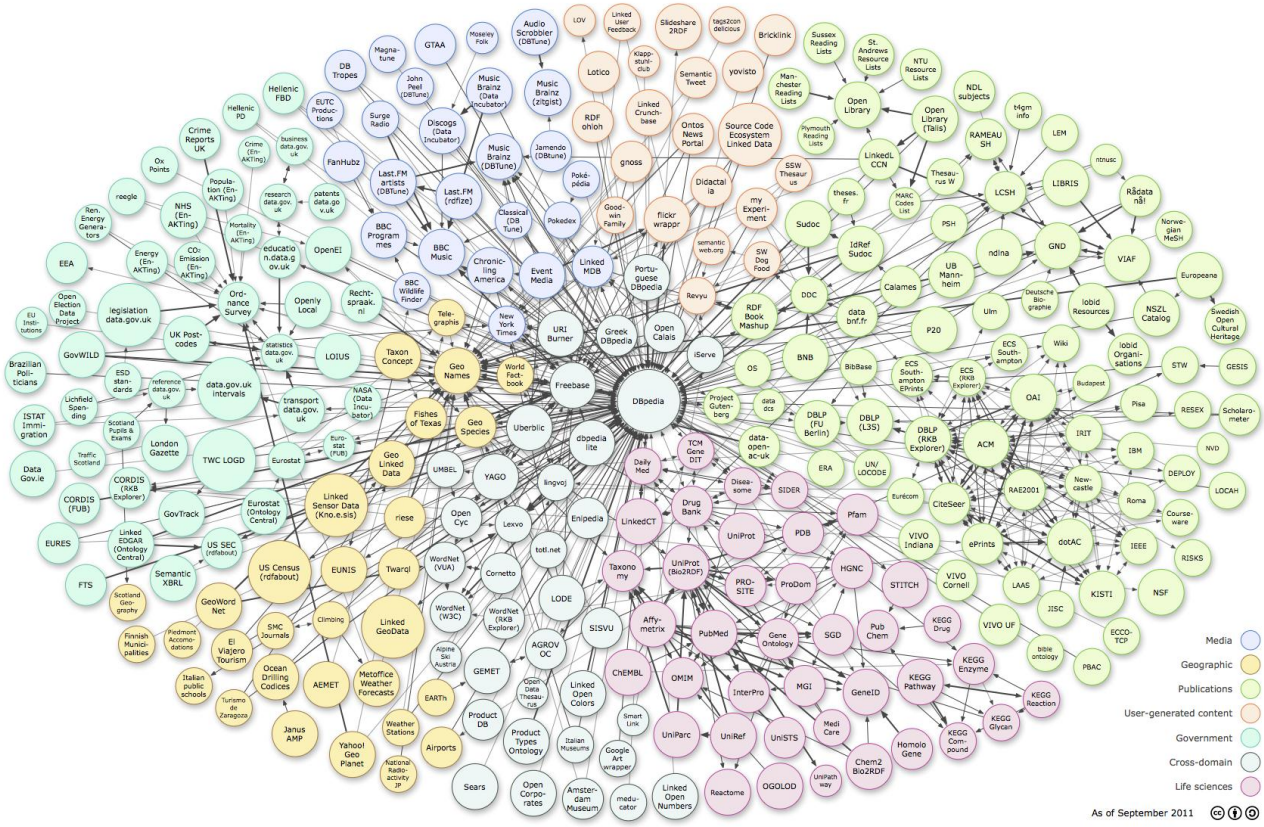

Figure 6 – Représentation graphique des ensembles de données dans le web de données dite *Linking Open Data Cloud* (Version de septembre 2011)

4.4 Les différents types de données du *Linked Data*

Le diagramme présenté en figure 6 fait apparaître de plusieurs couleurs les différents types de données présents actuellement dans le web de données. Il montre ainsi la diversité et la richesse des domaines couverts, pour une initiative relativement récente, mais aussi les domaines encore absents.

• *Les ressources d'intérêt général* (en bleu clair) recouvrent essentiellement les données issues de dictionnaires ou d'encyclopédies. De ce point de vue, le projet le plus emblématique est DBepdia, que nous avons déjà évoqué. Initiative lancée en 2007, DBpedia vise à extraire les informations structurées de Wikipedia et à rendre cette information disponible avec les technologies du web sémantique. Pour ce faire, DBpedia s'appuie sur les informations structurées déjà présentes dans l'encyclopédie collaborative :
- les « infobox », affichées dans un encart généralement présent à droite d'un article de Wikipedia, qui constitue une « carte d'identité » de la ressource décrite ;
- les liens reliant les différentes versions de la Wikipedia ;
- les catégories ;
- les liens présents dans l'article ;
- etc.

Mis au point et maintenu par l'Université de Leipzig, l'Université libre de Berlin et différentes sociétés commerciales, ce projet met à disposition un milliard de triplets RDF sur 3,64 millions de ressources, dont 416 000 personnes, 526 000 lieux, 106 000 albums musicaux...

• *Les ressources issues du « web social »* (en saumon) recouvrent les projets de conversion des services web existants, des sites web 2.0 aux technologies du web sémantique, l'exposition des données personnelles en utilisant le vocabulaire FOAF ou l'exposition des sites web « sociaux » (forums, blog, wikis, etc.) avec le vocabulaire SIOC.

• *Les ressources géographiques et touristiques* (en jaune) recouvrent les projets d'exposition de données géographiques et les projets de mise à disposition des données publiques dont une bonne partie est constituée de données statistiques. Parmi les projets de mise à disposition des ressources géographiques, on peut citer Geonames, système d'information géographique sous licence libre (CC BY), qui référence et donne les coordonnées géographiques de 8 millions d'emplacements, ou LinkedGeoData qui représente pour OpenStreetMap ce que DBpedia est à Wikipedia et qui contient 320 millions de points géoréférencés et 25 millions d'itinéraires.

• *Les données gouvernementales* (en bleu pervenche), auxquelles le mouvement d'accès ouvert a été engagé après l'annonce de la mise à disposition des données américaines par Barack Obama, à la suite de quoi a été développé le site data.gov. Si la première version de celui-ci n'intégrait pas les principes du web de données, la seconde version, disponible depuis mai 2010, a profité des avancées en la matière de son « cousin » britannique data.gov.uk, dirigé par Nigel Shadbolt et Tim Berners-Lee, ceux-ci ayant évidemment mis en pratique, pour le construire, leurs recherches dans le domaine.

• *Les ressources multimédia* (en bleu foncé) recouvrent des conversions de bases de données musicales en ligne comme Music Brainz ou Jamendo, mais aussi des initiatives plus originales comme celles de la BBC. Cherchant à valoriser et à mettre à disposition, dans une logique d'ouverture, les données accumulées depuis de nombreuses années, la BBC s'est tournée très rapidement vers les technologies du web sémantique. L'originalité de la démarche réside dans la réutilisation de données existantes dans le *Linked Data*. Ces données sont enrichies par les données produites par la BBC, afin de construire, à destination des utilisateurs, des sites web conviviaux qui présentent également l'avantage d'être manipulables par les machines. De ce point de vue, le site BBC Music constitue une réussite et un exemple précurseur pour la mise à disposition de données culturelles ou patrimoniales.

• *Les ressources médicales et biologiques* (en violet) recouvrent tous les ensembles de données qui ont été agrégés par le projet Bio2RDF et le groupe d'intérêt *Semantic Web Health Care and Life Sciences* (HCLS) du W3C. En effet, le modèle de graphes constitue le modèle de référence pour échanger les données biologiques réparties sur le réseau. Leur mise à disposition peut être cruciale pour accélérer la recherche dans la découverte d'un remède ou d'un vaccin pour telle ou telle maladie. Avec cet ensemble, le domaine de la biologie médicale démontre tout l'intérêt scientifique que revêt l'accès ouvert aux données brutes de la recherche.

• Enfin, *les données bibliographiques* (en vert) recouvrent à la fois des catalogues de bibliothèques comme Libris ou le SUDOC, des bibliographies sélectives type DBLP (bibliographie en informatique) ou *Semantic Web Dog Food* (bibliographie de différentes conférences dans le domaine du web sémantique), et des conversions selon les principes du *Linked Data* de services web existants (Amazon par exemple) comme RDF Book Mashup.

Les bibliothèques ont effectué un important travail pour mettre à disposition en particulier les fichiers d'autorité, jugés comme la catégorie d'information la plus susceptible de présenter un intérêt de réutilisation pour une grande diversité d'acteurs. Des initiatives localisées comme celle de la Library of Congress avec les LCSH, de la Bibliothèque nationale allemande avec ses *Gemeinsame Normdatei* (GND) ou celle de l'ABES avec idRef sont complétées par des projets d'envergure internationale comme VIAF, le fichier d'autorité international virtuel. La British Library a été la première à exposer dans le web de données sa bibliographie nationale. Enfin, le jeune data.bnf.fr de la Bibliothèque nationale de France présente l'intérêt de combiner une approche *Linked Open Data* avec une interface utilisateurs destinée à améliorer, entre autres, le référencement par les moteurs de recherche.

4.5 La réutilisation des données

L'intérêt de l'exposition des données dans le web de données ne réside pas seulement dans la navigation en parcourant les liens d'un jeu de données à un autre, mais dans la réutilisation de ces données pour créer de nouvelles applications. En effet, le principe du graphe global et la création de liens (équivalences de type <owl:sameAs> ou autres) facilitent l'agrégation de données provenant de différentes sources.

Cet atout est décrit de la manière suivante dans le rapport du groupe d'incubation du W3C *Bibliothèques et web de données* : « Dans l'écosystème actuel fondé sur les documents, les données sont toujours échangées sous la forme de notices, chacune d'entre elles étant supposée constituer une description complète. À l'inverse, dans un écosystème à base de graphes, une institution peut fournir un certain nombre de déclarations sur une ressource ; toutes les déclarations ainsi fournies sur une ressource donnée, identifiée de manière unique, peuvent alors être agrégées en un graphe global. On peut imaginer par exemple qu'une bibliothèque fournisse le numéro de bibliographie nationale d'une ressource et qu'une autre fournisse un titre traduit. Les bibliothèques pourraient employer ces déclarations provenant de sources extérieures de la même manière qu'elles le font aujourd'hui quand elles intègrent des images de couvertures de livres. Dans un écosystème de données liées, "il n'y a pas de petite contribution" – car toute contribution, aussi infime soit-elle, rend possible l'établissement de connexions essentielles, à partir de sources jusque là inconnues. [8] »

Ainsi, l'une des applications possible consisterait, pour notre Musée imaginaire Eugène Delacroix, à enrichir son propre site web d'informations qu'il n'aurait pas besoin de saisir lui-même dans sa base de données mais qu'il pourrait réutiliser de sources existantes comme DBpedia, VIAF, data.bnf.fr, Freebase, idRef, etc. Cela permettrait de fournir des biographies de personnes, des données sur les œuvres, des résumés, des traductions en différentes langues... autant de données qui ne sont pas disponibles dans la base du musée lui-même et seraient trop coûteuses à recréer pour que cela puisse être envisagé.

Le fait que toutes ces données soient déjà reliées et disponibles dans un formalisme commun facilite considérablement leur exploitation conjointe, même si celle-ci n'est pas totalement anodine : elle requiert en effet, outre l'alignement des URIs qui identifient de manières différentes des entités similaires, des analyses et traitements de données pour vérifier quelle information est disponible, comment elle a été modélisée, quelles sont les ontologies utilisées, comment celles-ci

peuvent être également mises en relation... Un fois ce travail d'analyse effectué, des traitements automatiques peuvent être mis en place pour rapprocher et manipuler les données, en utilisant les potentialités offertes par SPARQL, par exemple.

Si le Musée imaginaire Eugène Delacroix entreprend d'exposer ses propres données, il devient dès lors possible pour un tiers de créer un nouveau service, un *mash-up*, en réutilisant ces données en même temps que d'autres. C'est ainsi que se réalise alors pleinement le potentiel de ces technologies, en autorisant l'émergence de réutilisations inattendues, qui n'étaient pas forcément prévues au moment de la création des données et de leur mise en ligne.

Références bibliographiques

[1] Dean ALLEMANG, James HENDLER. *Semantic Web for the Working Ontologist: Effective Modeling in RDFS and OWL*. San Francisco : Morgan Kaufmann, 2008

[2] Muriel AMAR, Bruno MENON, coord. « Web sémantique, web de données : quelle nouvelle donne ? ». *Documentaliste Sciences de l'information*, décembre 2011. vol. 48, n° , p. 20-61

[3] Tim BERNERS-LEE. *An attempt to give a high-level plan of the architecture of the Semantic WWW*. Septembre 1998. http://www.w3.org/DesignIssues/Semantic.html

[4] Tim BERNERS-LEE. *Linked data*. Juillet 2006. http://www.w3.org/DesignIssues/LinkedData.html

[5] Fabien GANDON, Catherine FARON-ZUCKER, Olivier CORBY. *Le Web sémantique : comment lier les données et les schémas sur le Web ?* Paris : Dunod, 2012

[6] Thomas R. GRUBER. « A translation approach to portable ontologies ». *Knowledge Acquisition*, 1993, vol. 5, n° 2, p. 199-220. http://tomgruber.org/writing/ontolingua-kaj-1993.htm

[7] Tom HEATH, Christian BIZER. *Web de données : méthodes et outils pour les données liées.* Trad. française Julien Plu. Montreuil, Pearson France, 2012. http://www.pearson.fr/livre/?GCOI=27440100179400

[8] W3C INCUBATOR GROUP REPORT. *Library Linked Data Incubator Group Final Report* [Rapport final du groupe d'incubation « Library Linked Data » du W3C]. Octobre 2011. http://www.w3.org/2005/Incubator/lld/XGR-lld-20111025

Références des normes

[9] *Resource Description Framework (RDF): Concepts and Abstract Syntax*. W3C Recommendation, 10 February 2004. http://www.w3.org/TR/rdf-concepts

[10] *RDF Primer*. W3C Recommendation, 10 February 2004. http://www.w3.org/TR/rdf-primer

[11] *RDF Vocabulary Description Language 1.0: RDF Schema.* W3C Recommendation, 10 February 2004. http://www.w3.org/TR/rdf-schema

[12] *RDF/XML Syntax Specification (Revised)*. W3C Recommendation, 10 February 2004. http://www.w3.org/TR/rdf-syntax-grammar

[13] *OWL Web Ontology Language Overview*. W3C Recommendation, 10 February 2004. http://www.w3.org/TR/owl-features

[14] *XML Schema Part 2 : Datatypes Second Edition*. W3C Recommendation, 28 October 2004. http://www.w3.org/TR/xmlschema-2

[15] *Architecture of the World Wide Web, Volume one*. W3C Recommendation, 15 December 2004. http://www.w3.org/TR/webarch

[16] *Uniform Resource Identifier (URI): Generic Synta*x. IETF Network Working Group, Request for Comments 3986, January 2005. http://www.ietf.org/rfc/rfc3986.txt

[17] *SPARQL Query Language for RDF.* W3C Recommendation, 15 January 2008. http://www.w3.org/TR/rdf-sparql-query

[18] *SPARQL Protocol for RDF*. W3C Recommendation, 15 January 2008. http://www.w3.org/TR/rdf-sparql-protocol

[19] *SPARQL Query Results XML Format*. W3C Recommendation, 15 January 2008. http://www.w3.org/TR/rdf-sparql-XMLres

[20] *OWL 2 Web Ontology Language Document Overview*. W3C Recommendation, 27 October 2009. http://www.w3.org/TR/owl2-overview

[21] *Turtle Terse RDF Triple Language*. W3C Working Draft, 09 August 2011. http://www.w3.org/TR/turtle

[22] *SPARQL 1.1 Overview*. W3C Working Draft, 01 May 2012. http://www.w3.org/TR/sparql11-overview

[23] *RDF 1.1 Concepts and Abstract Syntax*. W3C Working Draft, 05 June 2012. http://www.w3.org/TR/rdf11-concepts

[24] *RDFa Core 1.1*. W3C Recommendation, 07 June 2012. http://www.w3.org/TR/rdfa-syntax

[25] *XHTML + RDF 1.1*. W3C Recommendation, 07 June 2012. http://www.w3.org/TR/xhtml-rdfa